



University  
of Glasgow

<https://theses.gla.ac.uk/>

Theses Digitisation:

<https://www.gla.ac.uk/myglasgow/research/enlighten/theses/digitisation/>

This is a digitised version of the original print thesis.

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study,  
without prior permission or charge

This work cannot be reproduced or quoted extensively from without first  
obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any  
format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author,  
title, awarding institution and date of the thesis must be given

Enlighten: Theses

<https://theses.gla.ac.uk/>  
[research-enlighten@glasgow.ac.uk](mailto:research-enlighten@glasgow.ac.uk)

# ANALYSIS OF SOME TEXTURED IMAGES BY TRANSPUTER

Xiaoling Leng

B. Sc. (Nanjing), M. Sc. (Manchester)

A thesis submitted to  
the University of Glasgow  
for the Degree of Doctor of Philosophy  
in the Faculty of Engineering

The University of Glasgow  
The Department of Civil Engineering

November 1992

ProQuest Number: 10992099

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10992099

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.

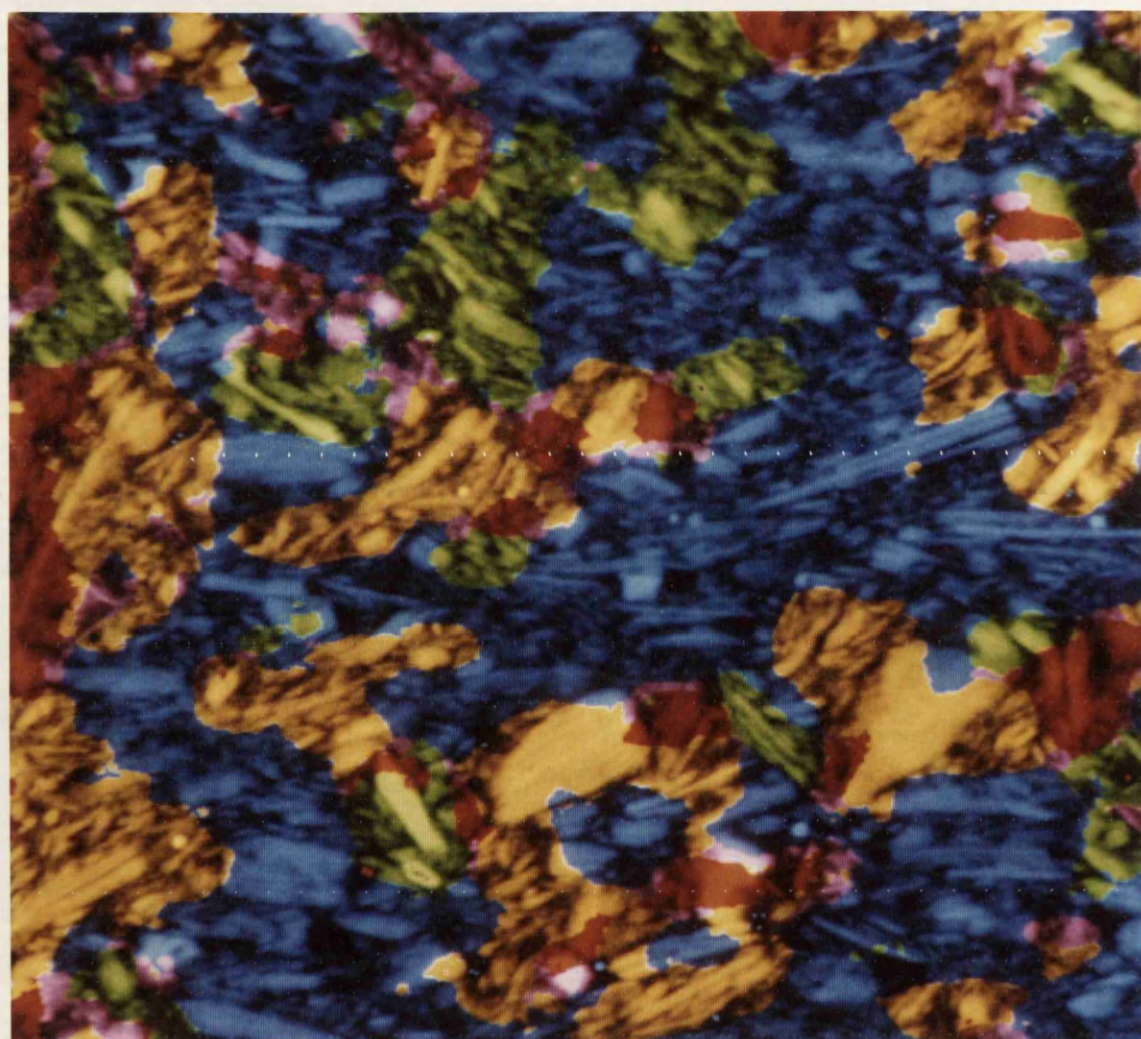
This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 – 1346

Thesis  
9410  
copy 1







**Frontispiece:** Scanning electron micrograph of a clay sample. This shows a cross-section of the sample, which is composed of platy particles, most of which are cut through by the cross-section, so they appear as lightly-coloured long thin objects. There are voids between the particles, which are dark. The micrograph has been mapped into groups of subparallel particles called domains and coloured as follow: blue, sub-horizontal; yellow, down-left; red, sub-vertical; green, down-right; purple, random cluster. In preparation for microscopy, the voids were impregnated with resin. The back-scattered mode of electron microscopy was used. The picture width is 20 microns. Same area as Fig. 3.1a.

To

my husband Hanxin Yang

my daughter Alice Manling Yang

## ACKNOWLEDGEMENTS

The work in this thesis was carried out whilst the author was a research assistant and research student in the Department of Civil Engineering, the University of Glasgow. It is my pleasure to thank all the staff and fellow students of the department for making my stay here both enjoyable and profitable. In particular, I would like to thank Professors D Muir Wood and D Green for giving the ideal facilities in the department for carrying out this research.

I wish to express my sincere thanks to Dr P Smart, who introduced and imparted to me the interesting field, Image Analysis by Parallel Processing, which is not only the very latest scientific development but also a brand new research area for me. During his supervision, he gave me much kind help, valuable training, instructive discussion and useful assistance in the preparation of this thesis.

Thanks are also due to Dr Derek Higgins for his skilled management of the University's MEIKO system; to the Scottish Transputer Centre, the SERC Transputer Loan Pool, Dr D Muir and his technicians in the Electronics and Electrical Engineering Department, Dr D White and his students in the Chemistry Department, and Miss J Sutherland for helping me in setting up the parallel systems.

Many thanks go to the following people for cooperation: Drs N K Tovey and M Hounslow in the University of East Anglia, Mr. L de F Costa in King's College of London University, Dr J MacLeod and Mr D Lou in the Electronics and Electrical Engineering Department of this University.

I would also like to acknowledge Dr I McConnochie for his kind help in applying for the SERC studentship and for carefully reading the thesis. Mr I Dickson, Mr A Better, other technicians, and all the secretaries in the department are thanked for providing friendly and useable facilities. Mrs Xiaohong Bai is acknowledged for providing some micrographs; Dr Y Ge is acknowledged for reading part of the thesis; Dr A Chan, Dr B Zhang, Mrs X Xue and other colleagues for encouragement and friendship.

My special thanks go to the American Airforce for supplying me with the research assistantship to cover the first two years expense, and to the British Science and Engineering Research Council for the research studentship to help me complete the research of this thesis.

Finally, I would like to thank my husband, Hanxin Yang, for continuous spiritual encouragement. Without his care and understanding, the work of this thesis could not have been completed.

## ABSTRACT

Texture, as a visual perception, can be easily seen by eye and often described without much difficulty. However, textural recognition and measurement by machine is a very different issue and has only recently been developed. In this thesis, a whole set of new algorithms have been developed to analyse textured images with particular reference to the requirements of soil microstructural applications. The new technology of parallel processing is used to implement and improve the complicated computations.

In this research, the techniques of Image Transformation, Image Segmentation, and Mathematical Morphology are used to carry out textural analyses of individual and groups of particles, and measurements taken over whole images. Rather than basing calculations directly on particle-by-particle calculations, as in most existing methods, these new approaches base the calculations on pixel-by-pixel calculations and analyse the vector given by the intensity gradient at each pixel.

Chapter 1 gives an introduction to the research.

Chapter 2 is a literature review of textural analysis.

In Chapter 3, two new textural segmentations, Top-Contouring and Consistency Ratio Mapping, are developed. These can be used to map areas in which most of the particles lie in the same direction and also random clusters, so as to analyse the texture of groups of particles. The application of these segmentations to other types of images such as aerial photographs was also successful.

Chapter 4 presents a new Orientation Analysis method, for the analysis of the global orientation distribution of micrographs based on directional data statistics. This method is compared with the Hough transform method, the Directed Vein method, and a Convex Hull method. These methods all produce polar histograms of the global orientation distribution which are in good agreement, and indicate the most popular directions of distribution similarly. A novel porosity analysis shows the relationship between the local porosity and local orientation characteristics in the image.

A Non-linear Greyscale Morphology algorithm has been developed and is discussed in Chapter 5. This new method can be applied directly to original grey level images rather than to the derived black-and-white images, thereby avoiding the deformation of features caused by binary conversion. The final result of successive cycles of erosion gives the skeletons of the particles. Further, these skeletons can be used to analyse the orientation, separation, and contact characteristics of individual particles in the image.

In order to accelerate the calculations of textural analysis, the latest computing development, parallel processing, was employed in the research. A review of parallel processing includes: level of parallelism; architectures and communications of parallel systems; parallel operating systems; and parallel languages, Chapter 6. The new algorithms were developed and implemented on parallel systems, Chapter 7. This involved setting up three multi-transputer systems, choosing a suitable parallel structure for the textural analysis, and developing the software for both the algorithms themselves and the harness required in the parallel systems.

Chapter 8 is a conclusion of the whole research and suggests some future work.

## RELEVANT PUBLICATIONS

The following papers, in the order in which they were relevant to the thesis, have resulted from this work:

"Textural Analysis" (Smart and Leng 1992a) is a review of textural analysis, which includes most of the information in Chapter 2.

"Automatic Analysis of Microstructure of Cohesive Sediments" (Smart et al 1989), "Improved Methods of Textural Analysis" (Leng and Smart 1991), "Image Analysis by Transputer" (Smart and Leng 1990), and "Textural Analysis by Transputer" (Smart and Leng 1991) mainly discussed the two new textural segmentations (Chapter 3) and the implementation of parallel processing for the calculations (Chapter 7). These resulted directly from the cooperation with the University of East Anglia.

"Semi-Automated Analysis of Clay Samples" (Costa et al 1991a), and "Analysis of Clay Microstructure by Transputer" (Costa et al 1991b) are the cooperation with King's College of London University on the application of the Hough transform on textural analysis (Section 4.3).

"Automation Orientation Analysis of Particles of Soil Microstructures" (Luo et al 1992) is the cooperation with the Department of Electronics and Electrical Engineering, which discussed the applications of the Directed Vein method and the Convex Hull method on Global Orientation Analysis and is covered in Section 4.4.

"Image Analysis of Clay Scanning Electron Micrographs" (Leng and Smart 1992) introduces the author's grey scale skeletonization



algorithm (Chapter 5).

"Present Developments in Image Analysis" (Smart and Leng 1992), and "Image Analysis of Soil Microstructure" (Smart et al 1992) are general reviews of the whole work.

## CONTENTS

FRONTISPIECE	II
DEDICATION	III
ACKNOWLEDGEMENTS	IV
ABSTRACT	VI
RELEVANT PUBLICATIONS	VIII
CONTENTS	X
LIST OF TABLES	XV
LIST OF FIGURES	XVI

CHAPTER 1 INTRODUCTION	1
1.1 PROJECT RATIONALE	1
1.1.1 Soil Microstructure and Microscopy	1
1.1.2 Quantitative Analysis and Previous Method	2
1.2 ALGORITHM DEVELOPMENT	3
1.2.1 Introducing Textural Analysis Theories	3
1.2.2 Improvement of Textural Analysis	4
1.3 PARALLEL PROCESSING FOR NEW ALGORITHM	5

### PART ONE: IMAGE ANALYSIS

CHAPTER 2 REVIEW OF TEXTURAL ANALYSIS	8
2.1 INTRODUCTION	8
2.2 METHODS	9
2.2.1 Clustering	9
2.2.2 Segmentation	10
2.2.3 Fourier Methods	10
2.2.4 Statistics	11
2.2.5 Structural Method	12
2.2.6 Large Filter Methods	13
2.3 ANALYSIS OF VARIANCE	15
2.3.1 Introduction	15
2.3.2 Mosaics	16

2.3.3 Surface Roughness . . . . .	16
2.3.4 Soil Surfaces . . . . .	17
2.3.5 Roads . . . . .	17
2.4 VARIOGRAMS . . . . .	18
2.4.1 Semi-variance . . . . .	18
2.4.2 Covariance . . . . .	19
2.4.3 Autocorrelation Coefficient . . . . .	20
2.4.4 Fractal Dimension . . . . .	20
2.5 DEPTH OF TEXTURE . . . . .	21
2.6 CONTRAST . . . . .	21
2.7 ACUTANCE . . . . .	22
2.8 SKEWNESS . . . . .	23
2.9 GRANULARITY . . . . .	24
2.10 SEPARATION AND DENSITY . . . . .	26
2.11 REGULARITY . . . . .	26
2.12 ENERGY . . . . .	27
2.13 ENTROPY . . . . .	27
2.14 POROSITY . . . . .	27
2.15 LENGTH AND AREA . . . . .	28
2.16 ORIENTATION . . . . .	28
2.16.1 Angular Cluster Analysis . . . . .	29
2.16.2 Morphological Method . . . . .	29
2.16.3 Hough Transform . . . . .	30
2.16.4 Intensity Gradient . . . . .	30
2.17 SUMMARY . . . . .	31
 <b>CHAPTER 3 TEXTURAL SEGMENTATION . . . . .</b>	 <b>34</b>
3.1 INTRODUCTION . . . . .	34
3.2 INTENSITY GRADIENT . . . . .	35
3.3 TOP-CONTOURING MAPPING . . . . .	37
3.3.1 Introduction . . . . .	37
3.3.2 Encoding the Intensity Gradient . . . . .	38
3.3.3 Smoothing the Encoded Image . . . . .	41
3.3.4 Investigation of Choosing the Best Radius . . . . .	49
3.4 CONSISTENCY RATIO MAPPING . . . . .	53
3.4.1 Introduction . . . . .	54
3.4.2 Summing Vectors . . . . .	56
3.4.3 Encoding Angles . . . . .	58
3.5 DISCUSSION . . . . .	60

<b>CHAPTER 4 GLOBAL ORIENTATION ANALYSIS</b>	<b>81</b>
4.1 INTRODUCTION	81
4.2 METHODS OF DIRECTIONAL DATA STATISTICS	82
4.2.1 Revealing Directional Information	82
4.2.2 Enhanced Orientation Information	84
4.2.3 Average Enhanced Vector and Consistency Ratio	86
4.2.4 Local Void Ratio	87
4.2.5 Enhanced Orientation Mapping	89
4.3 HOUGH TRANSFORM TECHNIQUE	90
4.3.1 Theory of Hough Transform	90
4.3.2 Hough Transform Application	92
4.4 DIRECTED VEIN AND CONVEX HULL	95
4.5 CONCLUSION	96

<b>CHAPTER 5 NON-LINEAR GREYSCALE MORPHOLOGY</b>	<b>110</b>
5.1 INTRODUCTION	110
5.2 BINARY MORPHOLOGY	111
5.3 PREVIOUS GREYSCALE MORPHOLOGY	116
5.4 NON-LINEAR STRUCTURE FUNCTION	118
5.5 THE CUBIC CURVE CORRECTION	122
5.6 LOOK-UP-TABLE AND PRACTICE	126
5.7 CONCLUSION	128

## PART TWO: PARALLEL PROCESSING

<b>CHAPTER 6 PARALLEL PROCESSING — PRINCIPLES</b>	<b>140</b>
6.1 INTRODUCTION	140
6.2 LEVEL OF PARALLELISM	140
6.3 ARCHITECTURE OF PARALLEL SYSTEM	143
6.4 COMMUNICATION WITHIN PARALLEL SYSTEM	146
6.5 PARALLEL OPERATING SYSTEM	147
6.6 PARALLEL LANGUAGE	149
6.7 APPLICATION OF TRANSPUTERS	149
<b>CHAPTER 7 PARALLEL PROCESSING — PRACTICE</b>	<b>155</b>
7.1 TRANSPUTER, TDS, AND OCCAM	155
7.2 HARDWARE SYSTEMS	159

7.3 SOFTWARE SYSTEMS . . . . .	.162
7.3.1 Introduction . . . . .	.162
7.3.2 Data-structure Parallelism . . . . .	.164
7.3.3 Communication . . . . .	.165
7.3.4 One Mapping Process . . . . .	.167
7.4 FUTURE DEVELOPMENTS . . . . .	.168
 CHAPTER 8 CONCLUSIONS AND FUTURE WORK . . . . .	 .187
8.1 CONCLUSIONS ON TEXTURAL ANALYSIS . . . . .	.187
8.1.1 Analysis of Domain Texture . . . . .	.187
8.1.2 Analysis of Global Texture . . . . .	.189
8.1.3 Analysis of Individual Particle Texture . . . . .	.190
8.1.4 Use of The Semi-variogram . . . . .	.191
8.2 CONCLUSIONS ON PARALLEL PROCESSING APPLICATION . . . . .	.191
8.3 SUMMARY . . . . .	.193
 APPENDIX A CHOOSING THE NUMBER OF TRANSPUTERS . . . . .	 .194
 APPENDIX B CODE FOR 'TEE' JUNCTION PROCESS . . . . .	 .196
 REFERENCES . . . . .	 .199

## LIST OF TABLES

Table 3.1 Typical <i>EXC</i> for radius=20 pixel and 4 direction. . . . .	62
Table 3.2 a) The number of pixels lying in different directions after different radii of Top-Contouring of Fig. 3.1a . . . . .	63
b) The difference from the average, after Tab. 3.2a	
Table 3.3 a) The number of pixels lying in different directions after different radii of Top-Contouring of Fig. 3.7 . . . . .	64
b) The difference from the average, after Tab. 3.3a	
Table 3.4 Distribution of the directions after Top- Contouring, and Consistency Ratio Mapping. . . . .	65
Table 3.5 Comparison of Top-Contouring and Consistency Ratio Mapping pixel by pixel. . . . .	65
Table 7.1 The transputer product range . . . . .	170
Table 7.2 Time (minutes) for Top-contouring mapping and display	171
Table 7.3 The calculations of swapping and filling for mapping 512*512 image onto four transputers cutting into either tiles and strips with 20 pixels radius filter . . . . .	172

## LIST OF FIGURES

Fig. 1.1	Parallelism of hardware and software must be designed in parallel . . . . .	6
Fig. 2.1	Perceptual texture dimension . . . . .	32
Fig. 2.2	Examples of texture histograms . . . . .	33
Fig. 3.1a	Scanning electron micrograph of soil at 2,000 magnification . . . . .	66
Fig. 3.1b	Differentiation of Fig 3.1a . . . . .	66
Fig. 3.1c	Encoded image of Fig. 3.1b. . . . .	67
Fig. 3.1d	Smoothed image of Fig. 3.1c . . . . .	67
Fig. 3.1e	Top-Contouring mapped image of Fig. 3.1a . . . . .	68
Fig. 3.2	Numbering of pixels within the mask . . . . .	68
Fig. 3.3	The process of Top-Contouring. . . . .	69
Fig. 3.4a	Encoding angles to 8 directions within $\pi$ . . . . .	70
	b Encoding angles to 4 direction within $2\pi$	
Fig. 3.5	a) Ellipse filter model for calculating <i>EXC</i> . . . . .	71
	b) Same ellipse filter model turned $90^\circ$ degree	
Fig. 3.6	Scanning electron micrograph with magnification 2000. 20 pixels radius has been chosen for Top-Contouring . . . . .	72
Fig. 3.7	Transmission electron micrograph with magnification 500, 35 pixels radius has been chosen for the Top-Contouring Mapping . . . . .	72
Fig. 3.8	a) Semi-variogram of Fig 3.1a . . . . .	73
	b) Scaled up semi-variogram	
Fig. 3.9	a) summed intensity gradient vector of Fig. 3.1a by Consistency Ratio Mapping . . . . .	74
	b) Segmented Fig. 3.1a by Consistency Ratio Mapping	
Fig. 3.10	The process of Consistency Ratio. . . . .	75
Fig. 3.11	Ultra-thin section of Kaolin on which the Top-Contouring mapping has subdivided horizontal and down-left domains from random areas . . . . .	76
Fig. 3.12	Comparison of two segmentations, with similar areas in grey, and the Top-Contouring colours in areas	

which differ for Fig. 3.1a . . . . .	76
Fig. 3.13 a) Horizontal domains of image in Fig. 3.10b . . . . .	77
b) Down-left domains of image in Fig. 3.10b . . . . .	
c) Vertical domains of image in Fig. 3.10b . . . . .	78
d) Down-right domains of image in Fig. 3.10b . . . . .	
Fig. 3.14 Consistency Ratio Mapping on small scale image . . . . .	79
Fig. 3.15 Consistency Ratio mapping on air photo of an unidentified island . . . . .	80
Fig. 4.1 Unitt's polar histogram shows the global orientation distribution of a scanning electron micrograph of clay. . . . .	98
Fig. 4.2 Backscattered scanning electron micrograph of sheared kaolin . . . . .	99
Fig. 4.3 Polar histograms calculated for Fig. 3.1a . . . . .	100
Fig. 4.4 a) Particle with sharp edges, . . . . .	101
b) histogram of angles for (a),	
c) particle with one diffuse edge,	
d) histogram of angles for (c).	
Fig. 4.5 Polar histograms calculated for Fig. 3.1a. . . . .	102
Fig. 4.6 Polar histogram of porosity of the image in Fig. 3.1a. . . . .	103
Fig. 4.7 a) Original image of a clay sample, . . . . .	104
b) detected edges of (a),	
c) detected straight line segments after Hough transform, (Coast 1991a).	
Fig. 4.8 a) x-y plane, (b) Hough a-b parameter space. . . . .	105
Fig. 4.9 a) Normal representation of a line, . . . . .	105
b) five point image,	
c) Hough $\rho$ - $\theta$ parameter space.	
Fig. 4.10 The polar histogram for the orientation distribution of the straight line segments in the image of Fig. 4.5a. . . . .	106
Fig. 4.11 Processing of the image corresponding to the domains with preferred orientation of $45^\circ$ . . . . .	107
Fig. 4.12 Example of Convex Hull of a particle. . . . .	108
Fig. 4.13 a) Experimental result by Directed Vein method. . . . .	109
b) Polar histogram of the directed Vein in Fig. 4.12a	
Fig. 5.1 a) The dilation of A by B . . . . .	112
b) The translation of A by x . . . . .	113



Fig. 5.2	a) The erosion of A by B . . . . .	.114
	b) The translation of A by $-x$	
Fig. 5.3	a) Binary transform of erosion . . . . .	.130
	b) Binary transform of dilation	
Fig. 5.4	a) Feature with unsymmetrical sides; . . . . .	.131
	b) feature has been cut by choosing grey level threshold of 200;	
	c) feature has been cut by choosing grey level threshold of 100;	
	d) transformed binary feature and its skeleton after process of (b);	
	e) transformed binary feature and its skeleton after process of (c).	
Fig. 5.5	a) Feature with unsymmetrical sides; . . . . .	.132
	b) feature has been cut by choosing grey level threshold of 200;	
	c) feature has been cut by choosing grey level threshold of 100;	
	d) transformed binary feature and its skeleton after process of (b);	
	e) transformed binary feature and its skeleton after process of (c).	
Fig. 5.6	Pixel N0 and its eight neighbours. . . . .	.117
Fig. 5.7	Dilation and erosion of a function $A(x)$ by a structure function $B(x)$ . . . . .	.133
Fig. 5.8	a) One cross section of the feature in Fig 5.4; . . . . .	.134
	b) gradient value at each point of (a);	
	c) result of eroded transform;	
	d) result of dilated transform.	
Fig. 5.9	Process of two-dimensional erosion on the feature in Fig. 5.4. . . . .	.135
Fig. 5.10	Cubic correction curves . . . . .	.136
Fig. 5.11	Erosion Look Up Table. . . . .	.137
Fig. 5.12	The process of skeletonization from a through b, c, d, to e. . . . .	.138
Fig. 6.1	The architecture of DAP 610. . . . .	.151
Fig. 6.2	Comparison of serial, pipelined and array structure . . . . .	.152
Fig. 6.3	Structure of pipeline processors. . . . .	.153
Fig. 6.4	Structure of array processors. . . . .	.154
	a) SIMD mode with fixed connections between	

processors and their local memory

b) MIMD mode with programmable connections between processors and memory.

Fig. 7.1	IMS T800 block diagram. . . . .	.173
Fig. 7.2	Basic structures formed by linking transputer together. . . . .	.174
Fig. 7.3	Wiring of a system of 5*T414 transputers on 2 boards in a RM Nimbus PC. . . . .	.175
Fig. 7.4	Wiring of a system of 6*T800 transputers on 3 boards . . . . .	.176
Fig. 7.5	INMOS Mother Board with 4 T800 TRAMs. . . . .	.177
Fig. 7.6	T800 TRAM with 2 MByte memory. . . . .	.177
Fig. 7.7	The PC showing the second monitor. . . . .	.178
Fig. 7.8	Overview interior of author's PC based multi-transputer system. . . . .	.178
Fig. 7.9	The links of author's multi-transputer system. . . . .	.179
Fig. 7.10	Wiring of a system of 6*T800 transputers (T6G domain) in Meiko computer. . . . .	.180
Fig. 7.11	The three basic stages of my image analysis, they can be regarded as three subtasks in parallel processing. . . . .	.181
Fig. 7.12	The arrangement of workspace in each working transputer . . . . .	.181
Fig. 7.13	Segmentation of an image onto four transputers. . . . .	.182
Fig. 7.14	Three processes on each working transputer running in parallel . . . . .	.183
Fig. 7.15	Four working processors array on multiprocessor block . . . . .	.184
Fig. 7.16	Simplified flow chart for Top-contouring mapping. . . . .	.185
Fig. 7.17	Illustrating the use of four transputer to Top-Contour a synthetic image consisting of a single circle. . . . .	.185
Fig. 7.18	Future parallel system structure. . . . .	.186

## CHAPTER 1 INTRODUCTION

### 1.1 PROJECT RATIONALE

#### 1.1.1 Soil Microstructure and Microscopy

Interest in extending soil mechanics study from macro-experiments to micro-structural analysis has recently become more important, since engineers have noticed that the mechanical characteristics of soils are anisotropic, and that microstructural changes affect the mode of soil consolidation and deformation. For example, when shearing consolidated clay, the particles may move around in groups. This movement will ultimately cause failure. However, how the particles move and how the failure actually occurs are still not well understood; and many engineers are very interested in learning more about these phenomena. Almost all that scientists know is that the characteristics of the movement will determine the place, the speed, and other aspects of the development of failure planes and the preceding deformation. Typical reviews are given by Smart and Tovey (1981), Bennett et al (1991), and O'Brien and Slatt (1990).

The Frontispiece shows a scanning electron micrograph of clay soil which is typical of those analysed by the author. The soil consists of small plate-like particles with voids between them. These voids were filled with resin, which hardened. Then the sample was sectioned and carefully ground flat. In the scanning electron

microscope, the particles are bright and the voids are dark. The colour was added subsequently as explained in Chapter 3.

Over the last decades, the means available to study the microstructure of soil have been improved. Optical microscopes, transmission electron microscopes, and scanning electron microscopes have all been applied to the analysis, thereby to see the details of soil microstructure. The results are useful in explaining the permeability, compressibility, strength and failure of the soil. Prior attempts were summarised by Smart and Tovey (1982). However, this field was only at its beginning and needed to be improved. By the time of starting the author's research, electron microscopy had been used for the analysis, and the backscattered electron technique had been recommended as a basic way to produce micrographs.

### **1.1.2 Quantitative Analysis**

For over 20 years, it had been necessary to use hand mapping and hand measurement on large scale electron micrographs to analyse the microstructure of clay soil. It takes a very long time and needs a skilled researcher to do this tedious and repetitive work. For example, an A4 print took about half a day of hand mapping and another half a day of hand measurement to analyse the orientation distribution; and even then much detailed information was left unmeasured. Therefore, improvement of quantitative analysis became unavoidable, and this required the introduction and application of computer and image analysis techniques to do the work automatically. The previous work done in this field included individual feature or domain measurement, overall orientation analysis, and void ratio studies. The intensity gradient concept and Consistency Ratio theorem

of the previous work have been adopted in the author's research.

Unitt (1975) was the first to use a microcomputer to analyse scanning electron micrographs of clay. He introduced the intensity gradient method to reveal the directional information of each pixel and analyse the anisotropy of the whole picture. A vector image calculated by an improved intensity gradient method (Smart and Tovey, 1980) became the basis for the algorithms developed in the author's research. The Consistency Ratio theorem, which had previously only been used in conjunction with hand mapping, has also taken an important place in the author's research, particularly with regard to Consistency Ratio Mapping and the associated algorithms.

## 1.2 ALGORITHM DEVELOPMENT

### 1.2.1 Introduction to Textural Analysis Theories

Image textures can be evaluated qualitatively as having one or more of the properties of fineness, coarseness, smoothness, granulation, randomness, lineation, or being mottled, irregular, or hummocky. Considering features in an image as particles or voids in a scanning electron micrograph of soil, these textural properties of an image given above can be used to describe the arrangement, the packing characteristics and the size of both particles and voids in micrographs, *i.e.* they describe the soil microstructure. So, the techniques of textural analysis could be employed to improve the quantitative study of soil microstructure.

In this thesis, the author reviewed the previous textural analysis algorithms and tried to import some suitable approaches for

the purpose of studying soil microstructure. The methods of textural analysis which have been introduced into this research include textural segmentation, filter smoothing, directional data statistics, the Hough transform, and the application of mathematical morphology. These are discussed in more detail in the next section.

### 1.2.2 Improvement of Textural Analysis

The previous work on textural analysis is summarized in Chapter 2 which covers textural segmentation, textural feature detection, and textural measurement. All these methods are based on the calculations of original grey level images; but what will be described later in this thesis are based on the calculations of a vector image, *i.e.* the intensity gradient at each pixel of the original image. So the new work brings the analysis to a vectorial calculation of textural segmentation, textural feature detection, and textural measurement.

Textural segmentation became the first stage of the research, because clay particles normally pack together parallelly and move in groups when deformation and shearing occurs. Two textural segmentations have been developed in Chapter 3 in order to analyse the behaviour of groups of particles. These two segmentations successfully map domains in which most particles lie in the same direction, and random clusters. After segmentation in this way, future measurement can include 'external' properties such as the sizes of the domain, and 'internal' properties such as the void ratio within individual domains. These measurements can then be related to different mechanical properties of the soil.

The second stage in the development of textural analysis improves

the measurement of the orientation characteristics of the whole image, so as to reveal information for a whole set of images and make comparisons between different soil samples. This new method shows the details of the anisotropic arrangement of particles in the image more clearly than the original intensity gradient method (Unitt 1976), and it draws attention on those directions in which many particles lie closely parallel (Chapter 4).

The third stage in the author's textural analysis development introduced a new algorithm of greyscale mathematical morphology, which avoids the deformation of features caused by binary conversion and produces the skeletons of particles or voids so as to analyse the further individual particle or void microstructure (Chapter 5). This provided the preparation for further work such as particle size arrangement, separation, contact, and orientation, much of which can be done using standard methods such as the Hough transform.

### 1.3 PARALLEL PROCESSING OF NEW ALGORITHM

For normal sequential computation, a new algorithm can be programmed directly. However, the limitation of the processor's speed and the associated memory prompted scientists to try parallel computing. For example, in the author's present research, a huge amount of information has to be analysed. Three soil mechanical experiments were being made, and 40 samples were obtained from each experiment with 25 images for each sample. A total number of 3000 images needed analysis. The size of each image is 0.25 Megabyte. Thus  $3000 \times 0.25$  MB together with other information, makes about 1000 MB of original data which needs very complex computation. Therefore, parallel processing was introduced into the research to improve the performance of the new

algorithms. Once the parallel system had been established, it was found straight forward to develop new algorithms into their parallel form when there is no reconfiguration<sup>u</sup> required.

Generally speaking, parallel processing tries to split either the computing tasks or the data onto different processors so as to perform calculations simultaneously. To do this there are three factors which need to be considered, namely parallelism, hardware, and software. Therefore, the purpose of the calculation and size of the data determine the method of splitting (mode of parallelism); and in turn, the mode of parallelism determines the method of the hardware linking. On the other hand, the present hardware set up also restricts the parallelism. Once the hardware is settled, the software has to follow the linking of the hardware, see Fig. 1.1.

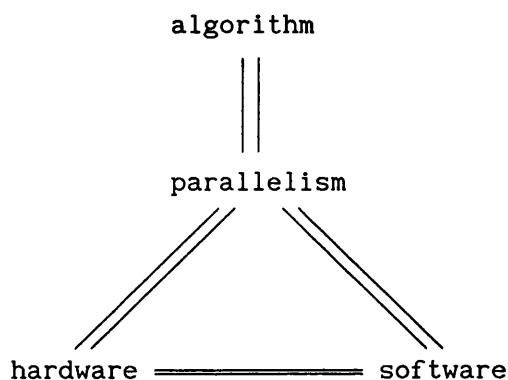


Fig. 1.1 Parallelism of hardware and software must be designed in parallel.

The software structure in the author's research was developed for the fullest possible use of the hardware facility and for the quickest possible developing and calculating of the new algorithms.

Before discussing these new developments, the next Chapter gives a review of textural analysis.



PART ONE IMAGE ANALYSIS

## CHAPTER 2 REVIEW OF TEXTURAL ANALYSIS

### 2.1 INTRODUCTION

*Definition: Texture is the net sum of those features which are too small to be worth analysing individually.*

Kaizer (1955) first tried to use a computer to quantify the visual perception -- texture of an image. Textural analysis became a serious subject of study in the late 1960s and early 1970s. Although there is no formal definition of texture in existence, textural analysis has been applied to many types of images such as medical images, aerial and satellite photographs, geological thin sections, and the scanning electron micrographs which were studied in this work.

The main objects of textural analysis are:

- 1) to discriminate between fields or features which are believed to be different;
- 2) to map features which differ from their surroundings;
- 3) to measure some property of the texture itself in the belief that this measurement can be interpreted directly.

Texture features are ubiquitously and obviously seen in the image and can be qualitatively evaluated as: fineness and coarseness; regularity and randomness; smoothness and roughness; granulation and mottle; density and separation; uniformity; frequency; orientation and

anisotropy; linearity; and so on. Referring to Levine (1985), a collection of artificial texture features are shown in Fig. 2.1.

Images are sometimes rectified and pre-processed before being analysed. The account given below assumes that any necessary rectification and pre-processing have been completed, and  $z$  will in general refer to the intensity of the resultant image. The possibilities include:

1. Differentiation: i.e.  $z = \text{mod grad } I$  or  $z = \text{grad } I$ .
2. Edge detection.
3. Skeletonisation.
4. Erosion to a point.

## 2.2 METHODS

Features which differ from their surroundings can be mapped using either cluster analysis or segmentation, as described below. A number of investigators have contented themselves with dividing the image into small tiles, but since large filters can be more accurate, this new method is also included in the review.

### 2.2.1 Clustering

Cluster analysis builds up features by placing into the same class pixels or small groups of pixels which appear to be similar. To give a simple example following Cook and Sandys-Renton (1991), we can map highly variable regions by the following procedure:

1. Divide the image into tiles of size  $8 \times 8$  pixels.
2. Let  $z$  be a suitable function of  $I$  within each tile.

3. Rank the tiles in descending order of  $\gamma$ .
4. Assuming for the sake of argument that the top 10 % of tiles in this ranking do belong to the same population, place these into a trial class  $H'$ , and test the next highest tile to see whether it belongs to the population defined by  $H'$ ; if so, add this tile to  $H'$ , and repeat this process.
5. When the next highest tile does not belong to  $H'$ , accept the latest version of  $H'$  as  $H$ , i.e. the class of highly variable tiles.

### 2.2.2 Segmentation

Segmentation cuts features out of an image by removing from consideration all pixels or small groups of pixels which fail to satisfy some criterion. In the example given above, all tiles with  $\gamma$  less than an arbitrary value  $\gamma_c$  are rejected and the remainder accepted as  $H$ .

### 2.2.3 Fourier Methods

Fourier transforms are ideal for analysing regular structures: the transform is a crystalline array of spots, from which the dominant orientations and wavelengths can be extracted. However: natural textured images are often irregular; and the spots broaden and merge to form a fuzzy and indecipherable mess. Perhaps the distribution of density within the transform could be analysed to provide useful results; but we have not seen a satisfactory study along these lines. Meanwhile, the best chances of success when using Fourier transforms appears to be to divide the image into small tiles of size  $8 \times 8$ ,  $16 \times 16$ , or  $32 \times 32$  pixels. (See Haralick, 1986.)

### 2.2.4 Statistics

The first-order probability distribution (or histogram) of the grey level within an image is a concise and simple summary of the information contained in the image. Suppose that  $z_k$  can take any number between  $\{0,1,\dots,maxgrey\}$  and the probability density function or first-order histogram is  $F(z_k)$ , then we have:

$$\text{mean:} \quad \mu = \sum_{z_k=1}^m z_k F(z_k) \quad (2.1)$$

$$\text{variance:} \quad \sigma^2 = \sum_{z_k=1}^m (z_k - \mu)^2 F(z_k) \quad (2.2)$$

$$\text{skewness:} \quad \mu_3 = \frac{1}{\sigma^3} \sum_{z_k=1}^m (z_k - \mu)^3 F(z_k) \quad (2.3)$$

$$\text{kurtosis:} \quad \mu_4 = \frac{1}{\sigma^4} \sum_{z_k=1}^m (z_k - \mu)^4 F(z_k) - 3 \quad (2.4)$$

$\mu$  and  $\sigma^2$  provide an indication of how uniform or regular a region is; skewness is a measure of how much the outliers in the histogram favour one side or another; kurtosis measures the effect of the outliers on the peak of the distribution, *i.e.* the degree of 'peakyness' (Levine 1985).

Second order statistics compare  $z(\mathbf{x})$  and  $z(\mathbf{x}+\mathbf{h})$ , where  $\mathbf{x}$  is the position of a pixel,  $\mathbf{h}$  is the separation between a pair of pixels known as the lag, and  $z(\mathbf{x})$  is the value of  $z$  at  $\mathbf{x}$ .  $\mathbf{x}$  and  $\mathbf{h}$  are vectors.

Haralick (1986) states that the co-occurrence  $F(a,b)$  of gray tones  $a$  and  $b$  for an image  $I$  is defined as the number of pairs of resolution cells (pixels) having grey levels  $a$  and  $b$ , respectively, and which are in a fixed spatial relationship, such as a fixed distance apart or a fixed distance and a fixed angle. Here:  $F(a,b)$  is the number of times and  $f(a,b)$  is the probability that  $g(\mathbf{x})=a$  and  $g(\mathbf{x}+\mathbf{h})=b$ ; and the Co-occurrence matrix is symmetrical. In this definition:

$$\sum_{a=0}^m \sum_{b=0}^m f(a,b) = 1 \quad (2.5).$$

Fourteen second order statistics have been worked out by Haralick et al (1973) to analyse the textural characteristics of an image.

Third order statistics, based on comparisons between three pixels,  $g(\mathbf{x})$ ,  $g(\mathbf{x}+\mathbf{h})$ , and  $g(\mathbf{x}+\mathbf{h}')$ , have also been used (see Levine, 1985).

### 2.2.5 Structural Method

The structural approach is normally used to analyse regularly textured images with well-defined primitives and pattern. Segmentation of texture primitives usually involves a prior selection of shape. The most obvious choice is an edge element; a second possible shape is a spot, which can be detected by using the characteristics of centre-surround receptive field. The latest method of edge detection in a binary image uses the Hough transform. Then these texture elements could be clustered into groups according to the shape attributes such as compactness, eccentricity, and orientation. The

approach is illustrated in Fig. 2.2 (Jacobus and Chein, 1981).

### 2.2.6 Large filter methods

The definition of *large* is elastic when applied to a filter, typically 20 pixels radius, containing about 1300 pixels. A *uniform* filter is one in which each pixel within the filter carries the same weight, so that, as the filter moves from one pixel to the next during its traverse of a line in the image, the trailing edge of the filter may be dropped and a new leading edge added. Thus, for 20 pixels radius, only 82 pixels around the perimeter are involved, a reduction of more than 90%. Uniform filters may be used when the algorithms are additive.

In the author's code, the shape of the filter is defined by the half chord lengths, e.g.  $RHS(P)$ , where  $P$  is the distance of the chord from the centre of the filter. Circular filters are used to minimise spurious orientation effects. The filter radius is specified in pixels, but 0.4 is added because this was found to improve the circularity of 'circles' up to 7 pixels radius. For a nominal radius of 1, the eight nearest neighbours are used; and for a nominal radius of 0, the four nearest neighbours are used. In the Fortran versions, the two sides of the filter are defined separately:

$$LHS(-P) = RHS(P) \quad (2.6),$$

so that shapes such as inclined ellipses or angular segments could be used if necessary. Square, rectangular, and straight-line filters are also possible.

For moving from line to line, the top edge is dropped and a new bottom edge is added. The top and bottom of the filter are defined by  $TOP(U)$  and  $BOT(U)$ , respectively, where  $U$  is the distance of the chord from the centre. A separate set of registers is maintained for the first pixel in each line, and it is these which are updated; then their contents are copied to the registers for the filter which traverses the line.

Three schemes are possible to avoid unnecessary tests within the double loop needed to process the whole image.

- a) In the first version of the author's code: the image array exactly equaled its nominal size; and the first filter on each line was set up without its leading edge. Then the operations within the loop were: add leading edge; calculate; drop trailing edge. A similar procedure was used for the 'leading column' filter. This method is difficult to code and to maintain.
- b) In the present code: the image array is extended by one line and by one column; (the extra pixels are set to zero for definiteness); and the first filter on each line is set up naturally. Then the operations within the loop are: calculate; drop trailing edge; add leading edge. At the last pixel in each line, the unnecessary leading edge is added, because this takes less time than would repeated testing to see whether the end has been reached. This is the simplest method to code.
- c) If the image array cannot be extended, then the loop could be shortened by one pixel, and the calculations for the last



pixel could be made outside the loop. This seems to demand the use of the INCLUDE extension of FORTRAN for the repeated portions of code; but this would perhaps make the development of new algorithms easier.

When using large filters, it is important to avoid leaving large unprocessed borders; in the author's code, the edge swapping and edge filling technique have been used. The image (512x512 size) is divided into four tiles of 256x256 size, Each quarter of the image is processed on one transputer with working space 512\*512 and 128 pixels wide border on each side. Swapping was used to process the inside borders, and reflecting was used to process the outside borders (details see Chapter 7, Smart and Leng 1990 and Smart et al. 1991).

It was difficult to fix the radius of the filter, because there was no predetermined optimum size for it. The only rule found was: to smooth away features of a given size, use a filter of a slightly larger size (Grant et al., 1990). The final choice of radius for the filter is discussed in Chapter 3.

## 2.3 ANALYSIS OF VARIANCE

### 2.3.1 Introduction

The region which is analysed may be: a series of images; an individual image; a series of similar features within one or more images; or an individual feature. Therefore, during the analysis, some statistical tests should be made along the lines sketched below.

### 2.3.2 Mosaics

If a series of images forms a mosaic of several series of features, and if the intensity,  $z$ , within each series of features is believed to have constant mean and variance throughout that series, then there is a danger that unsuspected interactions may have crept into the data. The experimental design is exactly that for which the classical analysis of variance was designed; and it is best to start by defining the statistical model of the data (e.g. see Snedecor, 1957). Obtaining the correct statistical model is even more important when using the variants described below.

### 2.3.3 Surface Roughness

In mechanical engineering, surface roughness is assessed as follow:

1. Select a number of sampling lengths.
2. Along each sampling length, measure the height above a suitable datum of the surface at equally spaced points.
3. Using least squares, fit a mean line through the data for each sampling length.
4. Hence find the profile departure,  $y$ , taken as the difference between the measured height and the corresponding height of the mean line.

Then, the root-mean-square deviation of the profile is given by:

$$R_q = \sqrt{\{ \sum y^2 / n \}} \quad (2.7),$$

where  $n$  is the number of sampling points (BS 6741: 1987). The standard

deviation of  $\psi$  is now less than that for the corresponding mosaic.

#### 2.3.4 Soil Surfaces

For an irregular surface such as that of soil, the profile departure,  $\psi$ , is better calculated from a moving average height, which is itself calculated as the average within a large filter (Grant et al., 1990). The standard deviation of  $\psi$  is now even less than that calculated by the previous method.

#### 2.3.5 Roads

The texture of roads is often measured by the sand patch method (BS 598: Part 105: 1990). This method is equivalent to: assuming that the original nominal surface was a ruled surface; measuring the profile departure,  $\psi$ , from this original; and calculating the mean value of  $\psi$ . As long as the original surface can be located, this method measures the amount of erosion of the surface; it could also be used to measure the amount of corrosion of a surface provided no complications arise. The standard deviation of  $\psi$  is now greater than that for the corresponding mosaic.

### 2.4 VARIOGRAMS

#### 2.4.1 Semi-variance

A *variogram* is a method of representing the spatial dependence in an image by considering the variability between pixels as a function of their separation. The variability is given by the semi-variance,  $g(h)$ , defined by:

$$g(h) = 1/2 E\{[\gamma(x) - \gamma(x+h)]^2\} \quad (2.8),$$

E denotes 'expected value of'.

Webster and Oliver (1990) calculated the semi-variance from:

$$g(+p, +q) = \frac{1}{2n} \sum_{i=1}^{l-p} \sum_{j=1}^{m-q} [\gamma(i, j) - \gamma(i+p, j+q)]^2 \quad (2.9a),$$

$$g(+p, -q) = \frac{1}{2n} \sum_{i=1}^{l-p} \sum_{j=1}^{m-q} [\gamma(i, j) - \gamma(i+p, j-q)]^2 \quad (2.9b),$$

$$g(-p, +q) = g(+p, -q) \quad (2.9c),$$

$$g(-p, -q) = g(+p, +q) \quad (2.9d),$$

where  $l, m$  is the picture size and  
where normally:

$$n = (l-p)(m-q) \quad (2.10),$$

except that any pairs missed from the calculation must be discounted, so it is safer to count  $n = n+1$  within the loop of the calculation. When calculating over a range of values of  $h$ , The author usually calculates only within a border of  $h_{\max}/2$  using pixels at:

$$i1 = i - p/2; \quad i2 = i1 + p; \quad \text{etc.} \quad (2.11).$$

The turning point of the Semi-variance graph can also be used to determine the filter size for smoothing in Top-Contouring and Consistency Ratio mapping, (see next Chapter, Fig. 3.8).

## 2.4.2 Covariance

Provided the mean,  $\mu$ , is constant throughout the region, then the covariance exists and is defined by:

$$\begin{aligned} c(h) &= E\{[z(x) - \mu][z(x+h) - \mu]\} \\ &= E[z(x) z(x+h)] - E z(x) \mu - E z(x+h) \mu + \mu^2 \quad (2.12), \end{aligned}$$

This may be rewritten as:

$$c(h) = E[z(x) z(x+h)] - \mu \mu - \mu \mu + \mu^2.$$

Therefore:

$$c(h) = E[z(x) z(x+h)] - \mu^2 \quad (2.13).$$

Putting  $h = 0$ :

$$c(0) = E\{[z(x)]^2\} - \mu^2 \quad (2.14),$$

$$\text{But: } g(h) = 1/2 E[z(x)^2 - 2 z(x) z(x+h) + z(x+h)^2]$$

$$= E\{[z(x)]^2\} - E[z(x) z(x+h)].$$

$$\text{Thus: } g(h) = c(0) - c(h) \quad (2.15),$$

i.e. the covariance, if it exists, is the mirror image of the semi-variance (Webster and Oliver 1990).

### 2.4.3 Autocorrelation Coefficient

If both the mean,  $\mu$ , and the standard deviation,  $\sigma$ , are constant throughout the region, then the autocorrelation coefficient exists and is defined by:

$$r(h) = c(h) / c(0) \quad (2.16).$$

$$\text{But now:} \quad c(0) = E\{[\gamma(x)]^2\} - \mu^2 = \sigma^2 \quad (2.17).$$

$$\text{Thus:} \quad r(h) = c(h) / \sigma^2 \quad (2.18),$$

$$\text{and:} \quad g(h) = \sigma^2 \{1 - r(h)\} \quad (2.19).$$

### 2.4.4 Fractal Dimension

An alternative approach, which is based on fractals, uses the graph of  $\log \sigma_y$  vs.  $\log \text{mod } h$ , where  $\sigma_y$  is the standard deviation of:

$$\psi = \gamma(x) - \gamma(x+h) \quad (2.20).$$

If this graph is a straight line: then the image is said to be self-similar at all scales; the fractal dimension,  $D$ , is given by:

$$D = 2 + G \quad (2.21),$$

where  $G$  is the gradient of the graph; and the self-similarity factor,  $H$ , is given by:

$$H = 1 - G \quad (2.22).$$

In practice, the graph is likely to be slightly curved. (Jardine and

Whitworth, 1991). Mandelbrot (1983) is a treasure house of further ideas.

## 2.5 DEPTH OF TEXTURE

The thought behind using the variance of  $\psi$  seems to be that it measures the depth of the texture. However, in mechanical engineering, the arithmetical mean deviation of the profile, which is given by:

$$R_a = \sum \text{mod } \psi / n \quad (2.23),$$

is more popular than  $R_q$  (Section 2.3.3). Alternatively, the differences in height between adjacent peaks and valleys may be measured and then averaged; or the histogram of  $\psi$  may be used to characterise the surface. (BS 6741: 1987).

## 2.6 CONTRAST

If it is accepted that a black-and-white image is the most contrasty possible, then when  $\gamma > \text{maxgrey}/2$ ,  $(\gamma - \text{maxgrey}/2)$  should be high; and when  $\gamma < \text{maxgrey}/2$ ,  $(\text{maxgrey}/2 - \gamma)$  should be high. On this basis, contrast is measured by:

$$C_1 = 4 \sum \gamma^2 / \{n m^2\} - 4 \sum \gamma / \{n m\} + 1 \quad (2.24),$$

where  $n$  is the number of pixels in the summations, and  $m$  is  $\text{maxgrey}$ . However, this is effectively using variance as a measure of contrast, and some investigators may prefer to use:

$$C_2 = \sqrt{(\sum \psi^4 / n)} \quad (2.25),$$

where  $y$  is the profile departure from the mean value of  $z$ . Note that  $C_1$  is based on  $z$  lying in its theoretical range of  $(0, \text{maxgrey})$ , whereas  $C_2$  depends on the actual range of  $z$ . Thus, a third approach is to use:

$$C_3 = \text{range}(z) \quad (2.26).$$

Haralick (1986) calls:

$$C_4 = \sum \sum (a - b)^2 f(a, b) \quad (2.27)$$

contrast, but this appears to be twice the semi-variance (see Section 2.4.1, the notation is in Section 2.2.4).

## 2.7 ACUTANCE

In photography, the sharpness of an edge is measured by its *acutance*,  $A$ , which is calculated from a traverse from bottom to top of the edge using:

$$A = \sum [\delta z / \delta x]^2 / [n \text{ range}(z)] \quad (2.28),$$

where the traverse is divided into equal intervals,  $\delta x$ , (Thomas, 1973). The value of  $A$  depends on where the traverse begins and ends. In mechanical engineering, the sharpness of edges is assessed from either the root mean square slope of the profile:

$$\delta_q = \sqrt{\{ \sum (\delta z / \delta x)^2 / n \}} \quad (2.29),$$



or the arithmetical mean slope of the profile:

$$\delta_a = \sum \text{mod } (\delta z / \delta x) / n \quad (2.30),$$

taken over several sampling lengths (BS 6741: 1987). The two-dimensional form of  $\delta_q$  is:

$$\delta_q = \sqrt{\{ \sum [(\delta z / \delta x_1)^2 + (\delta z / \delta x_2)^2] / n \}} \quad (2.31),$$

where  $x_1$  and  $x_2$  are image coordinates. An alternative is to use:

$$\delta_q' = \sum \text{mod grad } z / n \quad (2.32),$$

where the sum is taken over edge pixels only.

## 2.8 SKEWNESS

The skewness of the profile, which gives a measure of its shape, is:

$$S_k = \sqrt{(\sum \psi^3 / n)} / R_q^3 \quad (2.33),$$

where  $\psi$  is the profile departure from the mean value of  $z$ .

A more detailed analysis follows from plotting the profile bearing length ratio,  $t_p$ , vs.  $\psi$ , where:

$$t_p = l_p / l \quad (2.34),$$

where  $l_p$  is the length cut off at height  $\psi$ . (BS 6741: 1987). This

method extends easily to distinguishing between concave and convex surfaces: if  $\psi$  is measured from the deepest part of the profile, then  $\int (\psi/\psi_{\max}) dt_p$  is less than 0.5 for well-behaved concave surfaces.

## 2.9 GRANULARITY

Here, *granularity* is taken to mean the size of the texture-forming features. In photography, the term is used for the standard deviation of densiometer measurements, (Thomas 1973).

If the image can be reduced to black-and-white, and if the small texture-forming features are separable, then the mean of their areas, i.e. (total area) / count, can be taken as a measure of granularity. If they are not separable, then it may be possible to use the mean chord sizes instead; there will be separate sets of measurements for black and white areas. If the chord sizes are anisotropic, then it is preferable to measure them from separate scans at, say, 10 degree intervals.

If the Fourier transform shows a dominant size, this may be taken as a measure of granularity.

The nugget variance may also be usable as a measure of fine-scale granularity; but this idea has not been tested in this work. Mottles, i.e. coarse-grained granularity, are shown by the appearance of two ranges in the correlogram (Thomas, 1973).

The root mean square wavelength of the profile is defined by:

$$L_q = 2 \pi R_q / \delta_q \quad (2.35),$$

and is said to give a measure of spacing of local peaks and local valleys which takes into account their relative amplitudes and individual spatial frequencies. Presumably, the average wavelength of the profile:

$$L_a = 2 \pi R_a / \delta_a \quad (2.36),$$

does the same. (BS 6741: 1987).

According to Selwyn's Law, if:

$$k = \sigma a \quad (2.37),$$

where  $\sigma$  is the standard deviation of the intensity within a large filter, and  $a$  is the area of the filter; then,  $k$  is constant for an unmottled image (Thomas, 1973). For a circular filter, this may be tested either by the straightness of the graph of  $\log \sigma$  vs.  $\log d$ , where  $d$  is the diameter of the filter, or by calculation.

Mottles are also shown from the graph of  $\log \varphi(v)$  vs.  $\log v$ , where  $v$  is the spatial frequency, and  $\varphi(v)$  is the Wiener spectrum:

$$\varphi(v) = 2 \int \varphi(t) \cos 2\pi vt \, dt \quad (2.38).$$

The graph is horizontal for plain images and slopes downwards for mottled ones. (Thomas, 1973.)

## 2.10 SEPARATION AND DENSITY

If the image can be reduced to black-and-white, and if the small texture-forming features are separable, then their separations can be analysed either directly or from the chord size distribution, or from a graph of count vs. dilation.

Three slightly different concepts are: the spacing of the profile irregularities,  $S_m$ , i.e. the distance between two successive white edges in a black-and-white image; the spacing of local peaks, i.e. the distance between two successive peaks in a traverse of a grey image; and the corresponding spacing of local troughs (BS 6741: 1987). Remember that the last two of these are rather susceptible to the type of noise to which digitised images are prone.

The profile peak density is defined as  $1/S_m$  (BS 6741: 1987); some analysts would perform a grey scale skeletonisation before measuring it. More importantly, erosion to a point leads directly to the value of density, which may be the best discriminator for patches of texture of anomalous size.

## 2.11 REGULARITY

Crystalline *regularity* is usually best sought by Fourier transforms. Other types of regularity may be evident from periodicities in the semi-variogram, steps in the graph of count vs. dilation, etc. Local analyses within small tiles, large filters, or pre-mapped areas, are advisable.

## 2.12 ENERGY

Energy is defined by:

$$\sum \sum f(a,b)^2 \quad (2.39),$$

(Haralick, 1986).

## 2.13 ENTROPY

Entropy is defined by:

$$\sum \sum f(a,b) \log_e f(a,b) \quad (2.40),$$

(see Haralick, 1986). However, to avoid  $\log_e 0$ , a small constant value is added, for example:

$$\sum \sum \{ f(a,b) \log_e [(1 + F(a,b)) / n] \} \quad (2.41),$$

where  $n = \sum \sum F(a,b)$ . Entropy was introduced to measure information; but it also increases with noise (Shannon and Weaver, 1949).

## 2.14 POROSITY

If the image can be reduced to black-and-white, and if the small texture-forming features are then separable, then their local porosity can be found using a large filter.

If the small texture-forming features are not separable, then their local porosity,  $P$ , can be found for each pixel from:

$$P = (I_s - I)/(I_s - I_v) \quad (2.42),$$

where  $I_s$  and  $I_v$  are values attributed to pure solids and pure voids, respectively. If necessary, a gamma correction should be performed before making this analysis.

A porosity calculation in conjunction with the vector analysis has been carried out in this work (see Chapter 4).

## 2.15 LENGTH AND AREA

For a linear traverse, the developed profile length is:

$$L_o = \sum \sqrt{(\delta x^2 + \delta y^2)} \quad (2.43),$$

(c.f. BS 6741: 1987); and a corresponding formula could be found for the slope area of a two-dimensional surface. These measures could be used directly in their own right. Alternatively, they could be calculated as functions of scale. One method would be to define  $\psi$  in terms of lag. Another method would be to first calculate the area for the original image; then reduce the number of pixels by amalgamating blocks of 2 x 2 pixels and recalculating; and then repeat the reduction process. The results would provide another route to fractals.

## 2.16 ORIENTATION

Anisotropy and preferred orientation of the image is often shown when measurements are taken along linear traverses in different directions, the principal exception being the fractal dimension (see

Section 2.4.4); these methods are taken as self-evident here.

Some types of images, such as the scanning electron micrographs analysed here, are composed mainly of small texture-forming features which are themselves elongated. These small linear features may occur singly, or they may form either sub-parallel groups, called domains, or random clusters (Chapter 3). If the small linear features have two-fold symmetry, then their directions are said to be non-directed, and their orientations are modular to base  $\pi$ ; on the other hand, linear features such as arrows have definite directions.

### 2.16.1 Angular Cluster Analysis

In some images, the small texture forming features are rounded, mainly distributed randomly, but sometimes occurring in rows; examples are: clumps of rushes in a field, the rows following imperfect drains; sand grains along partings in geological materials; spots along the flanks of fishes. In such cases, it may be possible to find the rows by angular cluster analysis, where the angles concerned are the orientations of pairs of features. This presupposes that each feature can be reduced to a point, either by erosion, or by using its centroid. Presumably, some limit on separation will be imposed.

### 2.16.2 Morphology Method

An alternative approach to the rows of features discussed above would be to seek them using successive dilation and erosion with angular segment filters; features within rows in the direction of the filter would join during dilation and then resist erosion.

Successive dilation and erosion using a linear filter could be used to explore the directional properties of an image containing elongated features. The facility for this has long been available on image analysers such as the Quantimet and Leitz-TAS. A new morphology approach based on greyscale image is expressed in Chapter 5.

### 2.16.3 Hough Transform

The directional properties of an image containing elongated features can be analysed by Hough transform of their edges (Costa et al 1991a, 1991b); both polar histograms of orientation and segmentation on the basis of local preferred orientation are possible (see Chapter 4). It would probably be better to use skeletons in preference to edges (Chapter 5). The Hough transform could also be used to improve the grey-scale skeletonization. This suggestion has been described in more detail in Chapter 5.

### 2.16.4 Intensity Gradient Methods

The intensity gradient at each pixel of the image is calculated by:

$$U = \text{grad } I \quad (2.44a)$$

$$U = \text{mod grad } I \quad (2.44b)$$

$$A = \text{arg grad } I \quad (2.44c)$$

A whole series of methods based on the intensity gradient, have been developed in the author's work to analyse images with orientated features. These methods include: Top-Contouring and Consistency Ratio to map domains with local preferred orientation and random areas (Chapter 3); and the mean Vector Method to reveal the orientation



---

information over the whole image (Chapter 4).

## 2.17 SUMMARY

Textural analysis is a very difficult and complicated subject in image processing. Up to now, there has been no agreement on the definition of texture, even though it is obviously seen in the image. However, it seems impossible to make a machine recognize all the textures which are very easy for the human eye to recognize. So far, there are many methods which have been developed as described above. In order to analyse the scanning electron micrographs of clay, in which hundreds of different sized long thin particles point in all directions, a special series of textural methods have been developed in the author's work to analyse the microstructure of groups of particles, whole pictures, special areas, and individual particles.

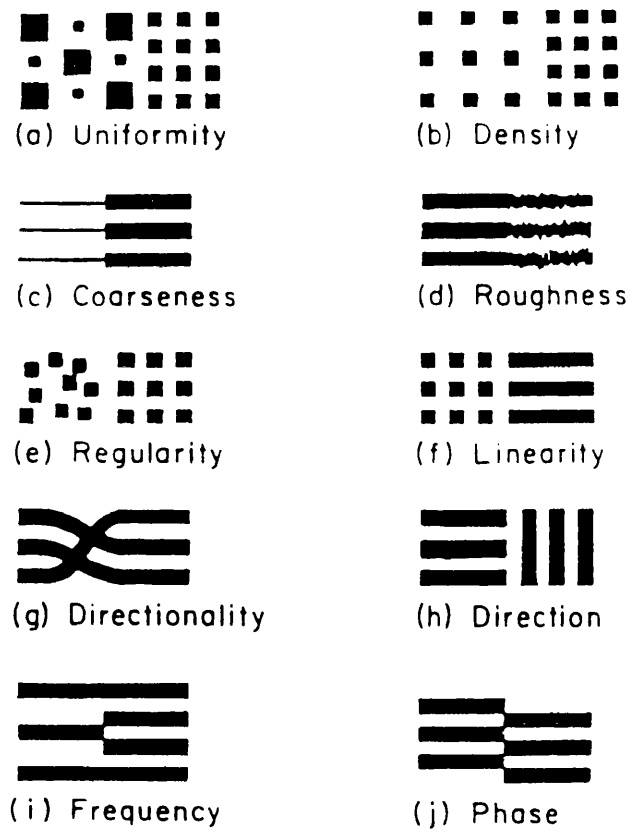


Fig. 2.1 Perceptual texture dimension. (Levine 1985)

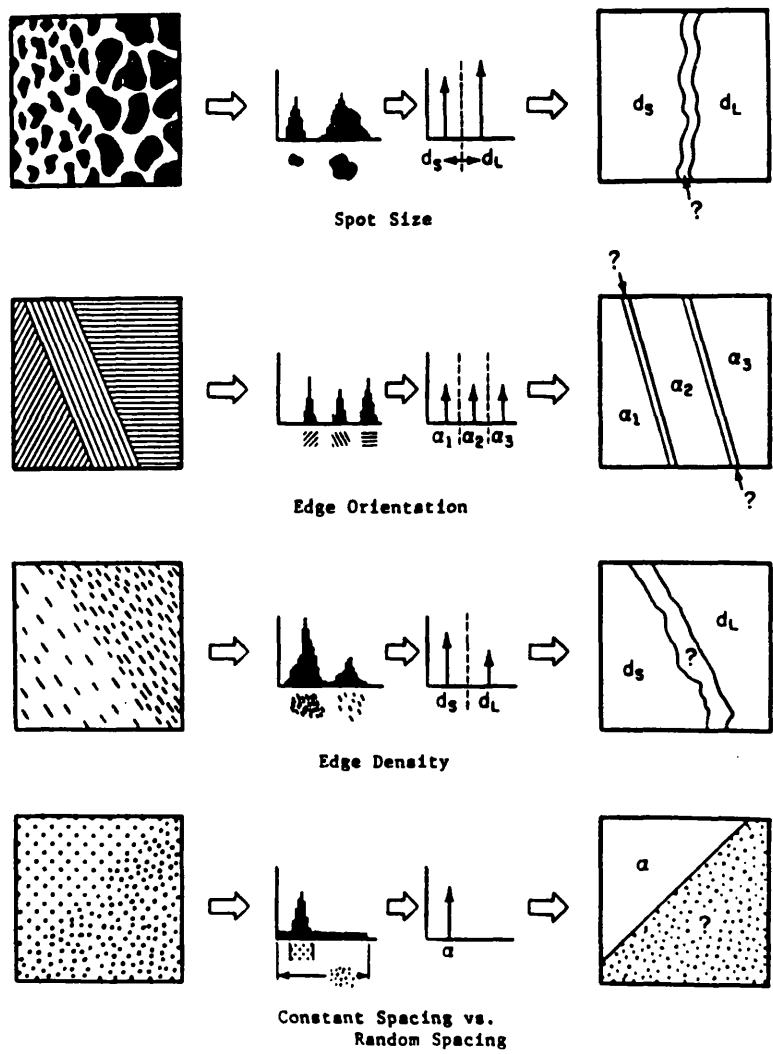


Fig. 2.2 Examples of texture histograms (Levine 1985)

## CHAPTER 3 TEXTURAL SEGMENTATION

### 3.1 INTRODUCTION

Clays are composed of micron and submicron-size particles which pack together either loosely or densely. There are several types of such particles; laths, tubes and, most commonly, 'thin plates', some of which have been described as 'slippery pennies'. The stable configuration for a pair of clay plates is face-to-face rather than edge-to-face, corner-to-edge, etc. There is, therefore, a strong tendency for the clay plates to be arranged in face-to-face groups (domains) with local preferred orientation (Fig. 3.1a).

In order to segment these domains and random areas, in addition to the methods described in the last Chapter, two new methods of image segmentation, Top-Contouring and Consistency Ratio Mapping, have been developed at Glasgow University. These have been used in the University of East Anglia to analyse two major experiments, and in turn they have been made available in the Semper Image Analysis System as standard Routines. Consistency Ratio Mapping is also being used by another research student in Glasgow to analyse a whole set of samples. The discussion of these two new segmentations is given in this Chapter. Because the main information on which the segmentations are based is the direction of the particles, the intensity gradient of each pixel in the image rather than the intensity itself has been employed as the fundamental element for developing these two new

segmentations.

### 3.2 INTENSITY GRADIENT

Unitt (1975) introduced the intensity gradient method to reveal some directional information on digitised scanning electron micrographs. In this method, a vector is calculated to indicate the greatest rate of change of intensity (i.e. of grey level) at each pixel in the image. For the case of a square raster, the intensity gradient vector  $U$  at point 0 was calculated from the four closest neighbour pixels, Fig. 3.2, viz.

$$U = \text{grad}(I) \quad (3.1)$$

$$U = \text{mod } (U) = \sqrt{X^2 + Y^2} \quad (3.2)$$

$$A = \text{arg } (U) = \arctan (Y/X) \quad (3.3)$$

where:

$$X = 1/2 (I_1 - I_3) \quad (3.4a)$$

$$Y = 1/2 (I_2 - I_4) \quad (3.4b)$$

where:  $I_1$  = intensity or grey level of the pixel at 1,

$X, Y$  = intensity differences in the  $x, y$  direction, respectively.

$U$  = strength of intensity gradient,

$A$  = angle of intensity gradient vector.

To improve the accuracy, Unitt (1976) used the one-dimensional Taylor expansion and the '9C' mask (Fig. 3.2 at 1,...4,9,...12). Later Smart (1981) suggested a more circular mask, using 17 pixels (Fig. 3.2 at 1,...12,21,...24). Smart and Tovey (1988) introduced the two-dimensional Taylor expansion to find the coefficients for

\* For each of the 20 points in the mask, an equation connecting the first 14 differential coefficients was written using the Taylor expansion; then a least-squares solution was used to find the differential coefficients in term of the intensities  $I(i,j)$  at the 20 points (together with  $I_0$  if needed).

different sizes of masks and different degrees of freedom within the 24 points shown in Fig. 3.2. Of the possibilities, the most circular '20' mask (Fig. 3.2 at 1,...20) and the most accurate '20-14-formula',\* have been used in the present research. The coefficient matrices for calculating the differentiation in the x-direction,  $dx_{ij}$ , and y-direction,  $dy_{ij}$ , are:

$$\begin{bmatrix} .0 & .013 & .0 & -.013 & .0 \\ .077 & -.207 & .0 & .207 & -.077 \\ -.070 & -.280 & .0 & .280 & .070 \\ .077 & -.207 & .0 & .207 & -.077 \\ .0 & .013 & .0 & -.013 & .0 \end{bmatrix} \quad (3.5)$$

$$\begin{bmatrix} .0 & .077 & -.070 & .077 & .0 \\ .013 & -.207 & -.280 & -.207 & .013 \\ .0 & .0 & .0 & .0 & .0 \\ -.013 & .207 & .280 & .207 & -.013 \\ .0 & -.077 & .070 & -.077 & .0 \end{bmatrix} \quad (4.6)$$

Then the strength of the two-dimensional intensity gradient can be calculated as:

$$X = \sum dx_{ij} I(i, j) \quad (3.7a)$$

$$Y = \sum dy_{ij} I(i, j) \quad (3.7b)$$

$$U = \sqrt{X^2 + Y^2} \quad (3.7c)$$

$I(i, j)$  are intensity at  $i, j$  position within mask. The angle of gradient vector equals:

$$A = \arctan \frac{\sum dy_{ij} I(i, j)}{\sum dx_{ij} I(i, j)} \quad (3.8)$$

In practice, if  $U$  is zero,  $A$  is indeterminate, and if  $U$  is very small, Eqn. (3.8) is inaccurate; thus  $A$  is labelled 'undecided'; if  $U$  is below some threshold value, see next section.

### 3.3 TOP-CONTOURING MAPPING

#### 3.3.1 Introduction

*Top-Contouring is a new textural segmentation which maps areas with local preferred orientation and random areas.*

The process of segmentation is:

- 1) calculating the intensity gradient of each pixel in the image, (Fig. 3.1b);
- 2) encoding the angle of the gradient vector into groups, (Fig. 3.1c);
- 3) smoothing the encoded angles by using a large circular filter, to get the local preferred orientation, (Fig. 3.1d);
- 4) mapping the pixels with the same preferred orientation into domains, (Fig. 3.1e).

The flowchart of the segmentation is given in Fig. 3.3.

Before encoding and smoothing, there are five parameters which need to be considered:

*NOD*: number of directions to which the particles in the analysed picture will be grouped;

*RAD*: the radius of the filter;

*LOA*: the lowest value of  $U$  for which  $A$  has decided direction.



*LOT*: the lowest acceptable number of pixels which lie within the filter and for which *A* has been decided, (if *LOT* is not attained, Top-Contouring Mapping labels the pixel as undecided);

*EXC*: the excess of the maximum number of pixels in a direction over the mean number of pixels per direction within the filter, which must be exceeded unless the pixel is to be labelled 'random' (here the direction with the maximum strength will be taken as the preferred local orientation);

### 3.3.2 Encoding the Intensity Gradient

The intensity gradient calculation of each pixel in the image has been described in Section 3.2. In the present *LOA* has been set as 2. Because *A* is calculated from Eqn. 3.8, if  $U=0$ , both  $dx$  and  $dy$  are zero, Eqn. 3.7 is indeterminate; and if both  $X$  and  $Y$  are small, the result may be dominated by errors in the digitization. To exclude these cases, *A* is labelled 'undecided' when  $U$  is small. so the angle of the vector may be calculated as:

If  $U \leq LOA$

then  $\alpha = 0$

If  $U > LOA$

then

If  $|y| > |x|$

then  $\alpha = \pi/2 - \arctan[|x| / |y|]$  (3.9a)

(which gives an angle of  $0 \rightarrow \frac{\pi}{4}$ )

If  $|y| \leq |x|$

then  $\alpha = \arctan[|y| / |x|]$  (3.9b)

(which gives an angle of  $\frac{\pi}{4} \rightarrow \frac{\pi}{2}$ )

This produces  $\alpha$  between 0 and  $\pi/2$ . The reason for using this method is for safety during division. Later the signs of  $y$  and  $x$  are used to ascribe  $\alpha$  to its proper quadrant.

In the analysis, because the two opposite directions belong to the same domain, the angle  $\alpha$  and  $\alpha+\pi$  will have the same code, therefore just the angles from 0 to  $\pi$  need to be considered.

$$\begin{array}{ll} \text{If } x \geq 0 \text{ AND } y \geq 0 & \\ \text{then } A = \alpha & (3.10a) \end{array}$$

$$\begin{array}{ll} \text{If } x < 0 \text{ AND } y \geq 0 & \\ \text{then } A = \pi - \alpha & (3.10b) \end{array}$$

$$\begin{array}{ll} \text{If } x < 0 \text{ AND } y < 0 & \\ \text{then } A = \alpha & (3.10c) \end{array}$$

$$\begin{array}{ll} \text{If } x \geq 0 \text{ AND } y < 0 & \\ \text{then } A = \pi - \alpha & (3.10d) \end{array}$$

In the author's early tests and often for display, the number of encoding directions  $NOD$  has been chosen as 4 (shown in Fig. 3.4b). Therefore:

Domain code	Direction	Angle $A$ ranges
0	undecided;	
1	horizontal	$0--1/8\pi$ , $7/8\pi--9/8\pi$ , $15/8\pi--2\pi$ ;
2	downleft	$1/8\pi--3/8\pi$ , $9/8\pi--11/8\pi$ ;
3	vertical	$3/8\pi--5/8\pi$ , $11/8\pi--13/8\pi$ ;
4	downright	$5/8\pi--7/8\pi$ , $13/8\pi--15/8\pi$ .
5	random	

On the other hand, the group in the University of East Anglia, for whom this method was developed, are using 8 or 12 encoding directions.

Next, the angles are encoded into  $2NOD$  directions within  $\pi$  (see Fig. 3.4a). The equation used here is:

$$cod = 1 + \{ \text{INT TRUNC} [ A * \frac{2 NOD}{\pi} ] \} \quad (3.11)$$

Then  $cod$  needs to be turned  $\pi/(2NOD)$  in the clockwise direction and then halved to match the  $NOD$  directions code as shown in Fig. 3.4b:

$$\text{If } cod \geq 2NOD \quad cod = 1$$

$$\text{At last } COD = 1 + [ cod / 2 ] \quad (3.12)$$

When the pixel has an undecided direction,  $COD = 0$ . The encoded image of Fig. 3.1a is shown in Fig. 3.1c, blue: sub-horizontal; yellow: down-left; red: sub-vertical; green: down-right; black: undecided.

### 3.3.3 Smoothing the Encoded Image

A large uniform circular filter is used in the smoothing process. A radius of 20 pixels has been chosen to analyse most of the images in this research. This section discusses the smoothing technique, preliminary details of the large filter method were given in Section 2.2.6. The next Section will give the details of choosing this best radius for smoothing.

Around each pixel, within the area defined by the filter, the smoothing process counts the number of pixels lying in each direction then finds the maximum value. It then carries out a test to see whether this is sufficiently high for the pixel to be labeled as having preferred orientation in that direction. Otherwise, this pixel will be labeled as random. Here the author will discuss the amount by which the strength of preferred orientation has to be exceeded over the average of the orientation distribution. First, several parameters could be defined as:

*MAX*: number of pixels lying in the most frequent direction within the filter;

*MEAN*: average number of pixels lying in each direction;

*MIN*: number of pixels lying in the least frequent direction in the filter;

*TOT*: total number of pixels which lie within the filter and have decided directions;

*LOX*: the limit which *MAX* must exceed unless the pixel is to be labelled 'random'.

According to the definition stated above:

$$MAX = MEAN + \frac{EXC}{100} \frac{TOT}{NOD} \quad (3.13a)$$

$$MIN = MEAN - \frac{EXC}{100} \frac{TOT}{NOD} \quad (3.13b)$$

$$\text{and: } MAX - MIN = 2 \frac{EXC}{100} \frac{TOT}{NOD} \quad (3.13c)$$

$$\frac{MAX - MIN}{TOT} = \frac{EXC}{100} * \frac{2}{NOD} \quad (3.13d)$$

Several filter models were considered to test the *EXC* parameter in this research. The first calculation was shown in SERC transputer report (Smart & Leng 1991), which was based on a function:

$$f(A,B) = A + B (\cos 2\theta) \quad (3.14)$$

as the distribution radius, therefore:

when	$\theta = 0$	$2\theta = 0$	$\cos 2\theta = 1$	$f = A + B$
	$\pi/2$	$\pi$	$-1$	$A - B$
	$\pi$	$2\pi$	$1$	$A + B$
	$3\pi/2$	$3\pi$	$-1$	$A - B$
	$2\pi$	$4\pi$	$1$	$A + B$

So

$$TOT = \int_0^{\pi} f \, d\theta$$

$$= \int_0^{\pi} (A + B (\cos 2\theta)) d\theta$$

$$= A\pi \quad (3.15)$$

$$MEAN = TOT / NOD = A\pi / NOD$$

The contributions to  $MAX$  are distributed around  $\theta=0$  within  $\pm\alpha$ , where  $\alpha = \pi/2NOD$ ,

$$\begin{aligned} MAX &= 2 \int_0^{\alpha} f \, d\theta \\ &= 2 \int_0^{\alpha} (A + B (\cos 2\theta)) d\theta \\ &= A \pi / NOD + B (\sin (\pi / NOD)) \end{aligned} \quad (3.16)$$

from (3.13a)

$$\begin{aligned} \frac{EXC}{100} &= \frac{MAX - MEAN}{MEAN} \\ &= \frac{A \pi / NOD + B \sin (\pi / NOD) - A \pi / NOD}{A \pi / NOD} \\ &= \frac{B}{A} \frac{NOD}{\pi} \sin (\pi / NOD) \end{aligned} \quad (3.17)$$

The summed distribution  $R$  can be calculated as:

$$R = \sqrt{X^2 + Y^2}$$

Where

$$\begin{aligned} X &= \int_0^{\pi} f (\cos 2\theta) \, d\theta \\ &= \int_0^{\pi} (A + B \cos 2\theta) (\cos 2\theta) \, d\theta \\ &= B\pi/2 \end{aligned} \quad (3.18)$$

$$\begin{aligned}
 Y &= \int_0^{\pi} f(\sin 2\theta) d\theta \\
 &= \int_0^{\pi} (A + B \cos 2\theta) (\sin 2\theta) d\theta \\
 &= 0
 \end{aligned} \tag{3.19}$$

$$\text{Therefore } R = X = B\pi/2 \tag{3.20}$$

According to the definition of Consistency Ratio (details see Section 3.4),

$$C = R / TOT \tag{3.21}$$

combines with eqns (3.20) and (3.15),

$$C = B / 2A$$

$$\text{i.e. } B/A = 2C$$

substitute in eqn. (3.17),

$$\frac{EXC}{100} = 2C \frac{NOD}{\pi} \sin(\pi/NOD) \tag{3.22}$$

and in terms of probability (see Smart and Tovey 1982),

$$\rho = e^{-C^2 TOT}$$

$$C = \sqrt{\frac{-\log_e \rho}{TOT}}$$

$$\frac{EXC}{100} = 2 \sqrt{\frac{-\log_e \rho}{TOT}} \frac{NOD}{\pi} \sin(\pi/NOD) \tag{3.23}$$

For  $NOD = 4$ , radius = 20,  $\rho = 0.01$ ,

$$EXC = 10.5\%$$

Several other filter models were also considered. As an example, an ellipse is discussed as a shape for the model. The ratio of long axis,  $a$ , to short axis,  $b$ , was defined as the measurement of the orientation distribution characteristics of the filter, *i.e.* the circularity of the ellipse measures the randomness of distribution. When  $a$  equals  $b$ , (a circle) the filter has random distribution. The ellipse is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (3.24)$$

The ellipse has a long axis of ' $2a$ ', with *MAX* distribution within  $\pm\pi/(2*NOD)$  around the long axis; and a short axis of ' $2b$ ' with *MIN* distribution within  $\pm\pi/(2*NOD)$  around the short axis (Fig. 3.5a). Due to the symmetry of the ellipse, only the range  $0^\circ - 90^\circ$  needs to be considered. The previous section has already discussed semicircular symmetry. So  $MAX = 2 \text{ OAC}$ , and  $MIN = 2 \text{ OBD}$  in Fig. 3.5a. For easy calculation, the area of OBD can be replaced by area of  $O'A'C'$  in Fig. 3.5b. So:

$$(MAX - MIN) / TOT = 4 (OAC - O'A'C') / \pi ab \quad (3.25)$$

and the formula for the ellipse in Fig. 3.5b is:

$$\frac{x^2}{b^2} + \frac{y^2}{a^2} = 1 \quad (3.26)$$

Lines OC and  $O'C'$  in Fig. 3.5 can be expressed as:

$$y = x \tan \alpha \quad (3.27)$$



Therefore:

$$\begin{aligned}
 (MAX - MIN)/TOT &= \frac{4}{\pi ab} \left\{ \int_0^{x_1} (x \tan \alpha) dx + \frac{b}{a} \int_{x_1}^a \sqrt{a^2 - x^2} dx \right. \\
 &\quad \left. - \int_0^{x_2} (x \tan \alpha) dx - \frac{a}{b} \int_{x_2}^b \sqrt{b^2 - x^2} dx \right\} \quad (3.28) \\
 &= \frac{4}{\pi ab} \left\{ \left[ \frac{1}{2} x^2 \tan \alpha \right]_{x_2}^{x_1} + \frac{b}{a} \left[ \frac{x}{2} \sqrt{a^2 - x^2} + \frac{a^2}{2} \arcsin \frac{x}{a} \right]_{x_1}^a \right. \\
 &\quad \left. - \frac{a}{b} \left[ \frac{x}{2} \sqrt{b^2 - x^2} + \frac{b^2}{2} \arcsin \frac{x}{b} \right]_{x_2}^b \right\}
 \end{aligned}$$

Now  $x_1$ , and  $x_2$  need to be calculated. Here  $x_1$  is the intercept of:

$$\begin{cases} y = x \tan \alpha \\ \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1; \end{cases}$$

So 
$$x_1 = ab / \sqrt{b^2 + a^2 \tan^2 \alpha}$$

similarly, 
$$x_2 = ab / \sqrt{a^2 + b^2 \tan^2 \alpha}$$

$$\begin{aligned}
 \text{Thus } (MAX - MIN)/TOT &= \frac{4}{\pi ab} \left\{ \frac{1}{2} \tan \alpha \left[ \frac{a^2 b^2}{b^2 + a^2 \tan^2 \alpha} - \frac{a^2 b^2}{a^2 + b^2 \tan^2 \alpha} \right] \right. \\
 &\quad + \frac{b}{a} \left[ \frac{\pi a^2}{4} - \frac{ab}{2 \sqrt{b^2 + a^2 \tan^2 \alpha}} \sqrt{a^2 - \frac{a^2 b^2}{b^2 + a^2 \tan^2 \alpha}} - \frac{a^2}{2} \arcsin \frac{x_1}{a} \right] \\
 &\quad \left. - \frac{a}{b} \left[ \frac{\pi b^2}{4} - \frac{ab}{2 \sqrt{a^2 + b^2 \tan^2 \alpha}} \sqrt{b^2 - \frac{a^2 b^2}{a^2 + b^2 \tan^2 \alpha}} - \frac{b^2}{2} \arcsin \frac{x_2}{b} \right] \right\}
 \end{aligned}$$

$$\text{so } (MAX - MIN)/TOT = \frac{4}{\pi ab} \left\{ \frac{ab}{2} \arcsin \frac{x_2}{b} - \frac{ab}{2} \arcsin \frac{x_1}{a} \right\} \quad (3.29)$$

$$= \frac{2}{\pi} \left\{ \arcsin \left[ \frac{x_2}{b} \sqrt{1 - \frac{x_1^2}{a^2}} - \frac{x_1}{a} \sqrt{1 - \frac{x_2^2}{b^2}} \right] \right\}$$

$$= \frac{2}{\pi} \left\{ \arcsin \frac{(a^2 - b^2) \tan \alpha}{\sqrt{a^2 + b^2 \tan^2 \alpha} \sqrt{b^2 + a^2 \tan^2 \alpha}} \right\} \quad (3.30)$$

$$= \frac{2}{\pi} \left\{ \arctan \frac{\frac{(a^2 - b^2) \tan \alpha}{\sqrt{a^2 + b^2 \tan^2 \alpha} \sqrt{b^2 + a^2 \tan^2 \alpha}}}{\sqrt{1 - \frac{(a^2 - b^2)^2 \tan^2 \alpha}{(a^2 + b^2 \tan^2 \alpha)(b^2 + a^2 \tan^2 \alpha)}}} \right\}$$

$$= \frac{2}{\pi} \left\{ \arctan \frac{(a^2 - b^2) \tan \alpha}{\sqrt{a^2 b^2 (1 + \tan^2 \alpha)^2}} \right\}$$

$$\text{therefore } \frac{MAX - MIN}{TOT} = \frac{2}{\pi} \arctan \frac{(a^2 - b^2) \tan \alpha}{ab \sec^2 \alpha}$$

$$= \frac{2}{\pi} \arctan \left[ \frac{a^2 - b^2}{2ab} \sin 2\alpha \right] \quad (3.31)$$

Combine with (3.13d)

$$\frac{EXC}{100} = [NOD/2] * \frac{2}{\pi} \arctan \left[ \frac{a^2 - b^2}{2ab} \sin 2\alpha \right] \quad (3.32)$$

In this example the number of directions for encoding has been chosen as 4. Therefore:

$$\begin{aligned}\frac{EXC}{100} &= \frac{4}{\pi} \arctan \left[ \frac{a^2 - b^2}{2ab} \sin 2\alpha \right] \\ &= 1.2732 \arctan \left[ \frac{a^2 - b^2}{2ab} \sin 2\alpha \right] \quad (3.33)\end{aligned}$$

From (3.33) it can be seen very clearly that the *EXC* value depends on the ratio of *b/a*, i.e. on the shape of the ellipse model. The calculated results of *EXC* are given in Table 3.1. The *EXC\** was also calculated by numerical integration using eqn. (3.13a)

$$EXC/100 = (MAX - MEAN) / MEAN$$

There is a little difference between the results calculated from these two methods. The table also shows the Consistency Ratio, *C*, and probability. In practice, when the distribution is unknown, it seems that either of these two distributions are reasonable and could be used to decide *EXC* as suggest above. Finally,

$$LOX = MEAN (1 + EXC/100)$$

The exact choice of *LOT* seems to be unimportant; and in the code it is set equal to *RAD*.

Then the statistics of the distribution of direction in the filter gives the number of pixels lying in the most frequent direction in the filter, *MAX*. The analysis is:

```
If  MAX < LOX
    then  TOP = NOD + 1          (random)
If  TOT < LOT
    then  TOP = 0                (undecided)
If  MAX = 0
    then  TOP = 0                (undecided)
```

otherwise

$TOP$  = the direction of  $MAX$

Thus the encoded number,  $COD$ , is replaced by the smoothed result  $TOP$  as shown in Fig. 3.1d.

### 3.3.4 Investigation of Choosing the Best Radius

The very important parameter which determines the pattern of mapping is the radius of the filter for smoothing. For large magnification micrographs, the area of each domain has to be big enough to contain the quite large particles; for small magnification micrographs, the domain has to be small enough without losing the fine structure. Sometimes the machine will divide very large particles into different parts according to the orientation of the intensity gradient around the sides of the particles (Fig. 3.7).

Several approaches have been considered in order to find the best choice of filter radius for mapping different micrographs. The first method was to judge the effect by eye. Then a statistical method was tried. Finally, a method based on the semi-variogram is suggested.

#### 1) Human Eyes Effect:

After Top-Contouring Mapping, the particles which have been ascribed to a certain domain are supposed to lie in a certain direction. However, high accuracy will bring the boundary of each domain as close as possible to each single particle. When the radius of the filter is equal to zero, the boundaries of the domains will be identical to the boundaries of the particles. This is not the purpose

of the smoothing. Top-Contouring is principally mapping domains, which can be considered to form a sub-parallel group with most particles lying in the same direction. Thus the boundaries of domains should not be drawn close to boundaries of single particles, so that they become very zigzagged. After comparing Top-Contouring Mapping on the same picture with different filter radii, the best radius for different pictures has been decided. Three examples are given below.

Fig. 3.6 is a micrograph with the magnification of 2,000, which has been used as the normal magnification by the University of East Anglia; the picture width is about 20  $\mu\text{m}$  and 512 pixel. The particle size at this magnification is reasonably large to show the detailed microstructure. At this ratio of particle and picture size, the best choice of radius as judged by the human eye was 20 pixels for the Top-Contouring Mapping.

Fig. 3.7 shows unconsolidated kaolin. The picture width is 512 pixel approximately 4  $\mu\text{m}$ . In order to show the microstructure more clearly, the magnification had been made much larger than that in Fig. 3.6, and it was included here to provide a wider basis of comparison. The particles within the picture are much larger than the particles in a normal micrograph; in particular, the particle which slopes down-right in the centre of the picture is nearly as long as the width of the picture. These large particles increased the best radius to 35 pixels for Top-Contouring smoothing, which is much larger than the best radius for a normal picture. The domain containing the central large particles has been mapped reasonably well. At the top of this domain, the particles are horizontal, and these have been distinguished correctly. At the lower left there is a feature, in a previous study it had been mapped by hand as a single vertical domain.

However it was divided into three regions by the Top-Contouring Mapping method , two of which were vertical and one inclined according to the edge intensity gradient of the particle.

There are both advantages and limitations in deciding the best radius by eye. For the human eye, the bigger and brighter particles attract more attention than smaller and fainter ones. When Top-Contouring Mapping was first developed, judgment by eye was inevitable, because objective methods had not then been developed. This, however, was entirely appropriate because the method had been developed from hand-mapping; but by now better methods have been developed.

## 2) Statistical Method

The first automatic method of choosing the best radius for Top-Contouring Mapping is a statistical method. This method is a modification of the chi-square error test on the histogram of the domains' areas with different radii choice  $H_R(k)$ , where  $R$  is the chosen radius and  $k$  is the direction. The observations are listed in Table 3.2a and 3.3a.

The average area of different radii is given (Table 3.2a and 3.3a) as:

$$\bar{H}(k) = 1/n \sum_R H_R(k) \quad (3.34)$$

where,  $n$  is the number of different radii which have been chosen in the test.

Then the difference between the observed area and the average area is presented in Tables 3.2b and 3.3b by using:

$$\Delta H_R(k) = \bar{H}(k) - H_R(k) \quad (3.35)$$

And the square sum of the difference in all the directions (Table 3.2b, and 3.3b) is:

$$\Delta H_R = \sum_k \Delta H_R(k)^2 = \sum_k [ \bar{H}(k) - H_R(k) ]^2 \quad (3.36)$$

So the radius with the minimum sum  $\Delta H_R$  could be the best radius of the filter for Top-Contouring Mapping. From Table 3.2b, the best radius for the micrograph with 2,000 magnification is 20 pixels. From Table 3.3b, the best radius for the micrograph of Fig 3.7 is 35 pixels. These agree reasonably well with the corresponding radii determined by eye. This analysis, which was made at an early stage in the project, gave some confidence in adopting  $RAD=20$  for 2000 magnification. However, the result is dependent to an extent on the details of the way in which the average areas have been calculated; so a better method was sought.

### 3) Semi-variogram Method

After Consistency-Ratio Mapping had been established (see next section), a geostatistical method, the semi-variogram, has been used to test the best radius. Fig. 3.8a is the semi-variogram of Fig. 3.1a, calculated for all lags up to 127 pixels in all directions. The brightness at each pixel is proportional to the semi-variance,  $g(h)$

defined by:

$$g(h) = \frac{1}{2} \sum [I(i) - I(i+h)]^2 \quad (3.37)$$

where  $i$ : and  $i+h$  are the positions of a pair of pixels,

$h$ : being the lag,

$I$ : the grey level.

The summation was taken over a large central portion of the micrograph. A circular model semi-variogram,  $f$ , was fitted to the data,  $f$  was given by:

$$f = c \left[ 1 - \frac{(t - \sin t)}{\pi} \right] \quad \text{when } h < a \quad (3.38)$$

$$f = c \quad \text{when } h > a$$

$$t = 2 \arccos (h/a) \quad (3.39)$$

where  $a$  and  $c$  are parameters, and  $t$  is introduced to simplify the formula (Webster and Oliver 1990).

The goodness of fit is illustrated in Fig. 3.8b. The semivariance had been scaled into arbitrary units to improve the visibility; but this is of no account here.  $a$  is known as the range and represents the limit of correlation between any pixel and its neighbours; it is therefore an appropriate upper limit on the size of a filter such as used for Top-Contouring, etc. The value of  $a$  obtained was 18 pixels, which agrees reasonably with the radii adopted above.

### 3.4 CONSISTENCY RATIO MAPPING

*Consistency Ratio Mapping is a textural segmentation which maps areas by summing the local vectors to find the preferred orientation.*



### 3.4.1 Introduction

Summing the intensity gradient vectors within a large filter gives the local preferred orientation, this segmentation, the Consistency Ratio Mapping, is more accurate than the Top-Contouring Mapping.

The Consistency Ratio itself is defined as a measure of the dispersion of the data, which is given by the normalized magnitude of the resultant vector, vector strength, or vector magnitude (Curaray 1956 and Smart and Tovey 1982 used unweighted vectors, but this work used weighted vectors). The definition are:

$$C = R / \sum U \quad (3.40)$$

$$\text{where } U_{ij} = \sqrt{X_{ij}^2 + Y_{ij}^2} \quad (3.41)$$

$$\text{and } R = \sqrt{Rx^2 + Ry^2} \quad (3.42)$$

$$Rx = \sum X_{ij} \quad Ry = \sum Y_{ij} \quad (3.43)$$

where  $X_{ij}$  and  $Y_{ij}$  are the intensity gradient at the (i,j) pixel in the x- and y-directions respectively.

The Consistency Ratio varies between zero, for a uniform distribution, and unity, for perfect preferred orientation (see Smart and Tovey 1982).

As mentioned above, Curray (1956) adapted a test to calculate the significance of two-dimensional data. If  $N$  is the number of vectors to be summed, then the probability,  $\rho$ , of obtaining a larger Consistency Ratio,  $C$ , by pure chance is

$$\rho = e^{-C^2 N} \quad (3.44)$$

There are several parameters which need to be defined before starting the calculation; they are

*RAD*: radius of the filter;

*NOD*: number of directions for segmentation;

*TOT*: total number of pixels which lie within the filter and which have decided directions;

*LOT*: the lowest acceptable number of pixels which lie within the filter and for which the intensity gradient has strength larger than zero, (if *LOT* is not attained, Consistency Ratio Mapping labels the pixel as undecided);

$R(i,j)$ : vector sum within the filter at pixel  $(i,j)$

$r$ : limiting value of  $R(i,j)$ , the value of summed vector could be labeled as random.

For the examples here, the author set:

$$RAD = 20$$

$$NOD = 4$$

$$LOT = RAD$$

and took  $N = TOT = 90\%$  of filter size for simplicity. To compare the

result of Consistency Ratio Mapping to that of Top-Contouring Mapping,  $\rho = 0.26\%$  has been chosen for the present analysis, because this brings these two results close to each other. Thus, from (3.44)

$$C = \sqrt{\frac{-\log_e \rho}{TOT}} \quad (3.45)$$

$$C = 0.07$$

and considering (3.40), the limiting value of the summed vector is

$$r = R = C \sum U_k \quad (3.46)$$

The whole process of Consistency Ratio Mapping segmentation is:

- 1) calculating the intensity gradient of each pixel in the image, (Fig. 3.1b);
- 2) summing up the vectors within the large circular filter;
- 3) encoding the angles of summed vectors to get the local preferred orientation, (Fig 3.9a);
- 4) mapping the pixels with the same preferred orientation into domains, (Fig. 3.9b).

The flow-chart of this process is given in Fig. 3.10.

### 3.4.2 Summing Vectors

Both the angle and the strength of the intensity gradient at each pixel are used to do the vector summing. Because the intensity gradient vectors surrounding each particle all point outwards, they are in opposite directions for two parallel edges, so the opposite

vectors cancel each other out. In order to remedy this problem, Krumbein (1939) suggested doubling the angles of the vectors before computation, and then halving afterwards.

From Section 3.2, the intensity gradient gives:

$$U(i, j) = \text{mod grad } I(i, j) \quad (3.47)$$

$$A(i, j) = \text{arg grad } I(i, j) \quad (3.48)$$

After doubling the angle, the values in the x-, and y-directions can be expressed as

$$X(i, j) = U(i, j) * \cos [2A(i, j)] \quad (3.49a)$$

$$Y(i, j) = U(i, j) * \sin [2A(i, j)] \quad (3.49b)$$

Then, summing up all the values in x-, y-direction within the filter:

$$R_x = \sum_k X(i, j) = \sum_k \{U(i, j) \cos [2A(i, j)] \} \quad (3.50a)$$

$$R_y = \sum_k Y(i, j) = \sum_k \{U(i, j) \sin [2A(i, j)] \} \quad (3.50b)$$

Where  $R_x$  and  $R_y$  are the components of the summed vector in x- and y-directions respectively.

Then the strength of the summed vector  $R$  equals:

$$R = \sqrt{R_x^2 + R_y^2} \quad (3.51)$$

The arithmetic sum of the strengths within the filter  $S$  equals:

$$S = \sum_k U(i, j) \quad (3.52)$$

### 3.4.3 Encoding Angles

Now the angle of the summed vector within the filter is calculated as:

$$\begin{aligned} \text{If } |Ry| \leq |Rx| \\ \text{then } \alpha = \arctan [ |Ry| / |Rx| ] \quad (3.53a) \\ \text{(which gives a angle } 0 \text{ -- } \frac{\pi}{4} \text{)} \end{aligned}$$

$$\begin{aligned} \text{If } |Ry| > |Rx| \\ \text{then } \alpha = \pi/2 - \arctan [ |Rx| / |Ry| ] \quad (3.53b) \\ \text{(which gives a angle } \frac{\pi}{4} \text{ -- } \frac{\pi}{2} \text{)} \end{aligned}$$

This first produced  $\alpha$  between 0 and  $\pi/2$ , after which the signs of  $Rx$  and  $Ry$  were used to ascribe  $\alpha$  to its proper quadrant.

$$\begin{aligned} \text{If } Rx \geq 0 \text{ AND } Ry \geq 0 \\ \text{then } A' = \alpha \quad (3.54a) \end{aligned}$$

$$\begin{aligned} \text{If } Rx < 0 \text{ AND } Ry \geq 0 \\ \text{then } A' = \pi - \alpha \quad (3.54b) \end{aligned}$$

$$\begin{aligned} \text{If } Rx < 0 \text{ AND } Ry < 0 \\ \text{then } A' = \pi + \alpha \quad (3.54c) \end{aligned}$$

$$\begin{aligned} \text{If } R_x \geq 0 \text{ AND } R_y < 0 \\ \text{then } A' = 2\pi - \alpha \end{aligned} \quad (3.54d)$$

which gives the angle of the summed vector from 0 to  $2\pi$ . The angle then needs to be halved, gives angle between 0 —  $\pi$ .

$$A = A' / 2 \quad (3.55)$$

When encoding, the angles of summed vectors encoded to  $2NOD$  directions within  $\pi$  first, (Fig. 3.4a).

$$\begin{aligned} \text{If } TOT < LOT \\ \text{then } con = 0 \quad \quad \quad (\text{undecided}) \end{aligned} \quad (3.56a)$$

$$\begin{aligned} \text{If } R < r \\ \text{then } con = NOD + 1 \quad \quad \quad (\text{random}) \end{aligned} \quad (3.56b)$$

Otherwise

$$\begin{aligned} con = 1 + \{ \text{INT TRUNC} [ A * 2 * NOD / \pi ] \} \\ (\text{encoded into } 2 * NOD \text{ within } \pi \text{ first}) \end{aligned} \quad (3.56c)$$

Then the  $con$  needs to be turned  $\pi/(2NOD)$  in the clockwise direction, and encoded to  $NOD$  directions (Fig. 3.4b):

$$\begin{aligned} \text{If } con \geq 2NOD \\ CON = 1 \end{aligned}$$

Otherwise

$$CON = 1 + [ con/2 ] \quad (3.57)$$

When a pixel has undecided direction,  $COD = 0$ .

Since the number of code directions *NOD* was chosen as 4, we have:

domain code	Direction	Degree range <i>CON</i> (i,j)
0	undecided	
1	horizontal	$0—1/8\pi$ , $7/8\pi—8/8\pi$ ;
2	downleft	$1/8\pi—3/8\pi$ ;
3	vertical	$3/8\pi—5/8\pi$ ;
4	downright	$5/8\pi—7/8\pi$ ;
5	random	

Fig 3.9 a was encoded result; then the boundaries were superimposed on the original, Fig. 3.9 b; finally the grey level image was coloured, Frontispiece.

### 3.5 DISCUSSION

Top-Contouring and Consistency Ratio Mapping are both based on the intensity gradient orientation information and map domains with local preferred orientation and random areas successfully. In the past, these features have been mapped by hand, but now these two new segmentations provide automatic ways to map domains and random clusters and the possibility of further measurements of features. Fig. 3.11 is an ultra-thin section of kaolin, which had originally been classified merely as random structure by hand (Smart et al 1991). However, the automatic mapping has subdivided the micrograph into approximately equal areas of horizontal, down-left, and random, with negligible areas of vertical and down-right. In this example, the quantitative instrumental method has retrieved information which had not been obtained by eye.

Consistency Ratio Mapping was only developed after the present work had been moved to the MEIKO machine, where each transputer has 4 megabyte memory (see Chapter 7). This extra memory enabled the author to consider the weighted gradient vector instead of just using the angle of the vector. There appear to be two reasons why Consistency Ratio Mapping should be more accurate than Top-Contouring Mapping. Firstly, Consistency Ratio Mapping uses the weighted intensity gradient; however, there is a counter argument that the weighting depresses the influence of small faint particles in comparison with larger, more intensive particles (Tovey 1991). Secondly, smoothing before encoding, as in Consistency Ratio Mapping, is expected to be more accurate than smoothing after encoding, as in Top-Contouring Mapping. Table 3.4 is the directional distribution of the image Fig. 3.1a by using both segmentations. Fig. 3.12 and Table 3.5 give the comparison between these two segmentations. The large values are in the positions when *TOP* and *CON* are in the same direction. There are some differences, however, the maximum difference is less than 3% of the whole image.

In preparation for further analysis, Figs. 3.13a, 3.13b, 3.13c, and 3.13d are the extracted images of Fig. 3.1a with preferred orientation in the horizontal, down-left, vertical, and down-right directions respectively. The methods can be very well employed for different scales of scanning and transmission electron micrographs, as can be seen in Figs. 3.1e, 3.6, 3.7, 3.9 b and 3.14. They can also be used for analysing other kinds of images, such as Fig. 3.15, an airphoto of an unidentified island. The segmentation successfully draws boundaries around areas within which all the mountains lie in certain directions. Postprocessing is now required to map the ocean.



Table 3.1 Typical *EXC* for radius = 20 and 4 directions

b/a	<i>EXC</i>	<i>EXC*</i>	<i>C.R.</i>	<i>PROB</i>
0.0000	0.0000	0.0000	0.0000	1.0000
0.0196	1.7829	1.7953	0.0099	0.8765
0.0385	3.5311	3.5800	0.0196	0.5962
0.0566	5.2460	5.3540	0.0219	0.3195
0.0741	6.9289	7.1172	0.0385	0.1367
0.0909	8.5809	8.8696	0.0476	0.0474
0.1071	10.2031	10.6110	0.0566	0.0134
0.1228	11.7966	12.3415	0.0654	0.0032
0.1379	13.3623	14.0609	0.0741	0.0006
0.1525	14.9013	15.7692	0.0826	0.0001
0.1667	16.4143	17.4664	0.0909	0.0000

here b/a = ratio of axes of the ellipse;

*EXC\** = approximate value calculated from ellipse model

*EXC* = value calculated by numerical integration using  
 $100 * (MAX - MEAN) / MEAN$ ;

*C.R.* = Consistency Ratio calculated by numerical integration.;

*PROB* = probability corresponding to *C.R.*, i.e.  $\exp(-C^2/N)$ , without  
any allowance for undecided pixels.

Table 3.2a The number of pixels lying in different directions after different radii of Top-Contouring of Fig. 3.1a

Coded directions						
$H_R(k)$ radii	0 undecide	1 horizont %	2 downleft	3 vertical	4 downright	5 random
0	3.8	19.5	20.4	30.3	25.6	0
5	0	10.8	11.3	35.2	39.9	2.5
10	0	5.2	9.43	40.4	43.9	0.9
15	0	2.7	7.6	43.51	45.8	0.2
20	0	1.3	5.6	46.1	46.7	0.1
25	0	0.4	3.1	48.3	47.9	0
30	0	0.1	1.5	48.7	49.6	0
35	0	0	0.8	48.6	50.4	0
40	0	0	0	48.9	51.0	0
45	0	0	0	47.9	52.0	0
50	0	0	0	47.6	52.3	0
60	0	0	0	46.0	53.9	0
128	0	0	0	32.5	67.4	0
average $\bar{H}(k)$	0.27	2.857	4.26	43.63	48.53	0.264

Table 3.2b The difference from the average, after Tab. 3.2a

$ \Delta H_R(k) $ radii	0	1	2	3	4	5	$\Delta H_R$
0	3.53	16.64	16.145	13.328	22.932	0.264	992.932
5	0.27	7.94	7.04	8.428	8.632	2.236	263.220
10	0.27	2.34	5.14	3.228	4.632	0.636	64.248
15	0.27	0.157	3.34	0.128	2.732	0.064	18.737
20	0.27	1.557	1.34	2.472	1.832	0.164	13.787
25	0.27	2.457	1.16	4.672	0.632	0.264	29.752
30	0.27	2.757	2.76	5.072	1.068	0.264	42.227
35	0.27	2.857	3.46	4.972	1.868	0.264	48.684
40	0.27	2.857	4.26	5.272	2.468	0.264	60.338
45	0.27	2.857	4.26	4.272	3.468	0.264	56.73
50	0.27	2.857	4.26	3.972	3.768	0.264	56.427
55	0.27	2.857	4.26	3.172	4.568	0.264	57.578
60	0.27	2.857	4.26	2.372	5.368	0.264	60.894
128	0.27	2.857	4.26	11.128	18.868	0.264	506.286

**Table 3.3a** The number of pixels lying in different directions after different radii of Top-Contouring of Fig. 3.7

$H_R(k)$ radii	Coded directions					
	0	1	2	3	4	5
	undecide	horizont	downleft ( % )	vertical	downright	random
0	9.7	19.9	26.9	25.8	17.4	0
5	1.1	22.2	22.9	31.2	20.0	2.2
10	0.3	25.3	23.2	34.4	15.7	0.9
15	0	26.3	24.1	35.2	13.8	0.3
20	0	26.0	25.2	35.6	12.8	0.1
25	0	26.2	26.5	35.2	11.8	0
30	0	25.6	27.5	35.4	11.1	0.1
35	0	25.1	29.2	35.1	10.3	0
40	0	25.9	30.4	34.3	9.2	0
45	0	25.5	32.4	33.0	8.9	0
50	0	23.9	34.5	33.6	7.8	0
55	0	22.6	36.1	34.6	6.5	0
60	0	20.2	38.0	34.9	6.7	0
128	0	6.4	47.7	45.7	0	0
average $\bar{H}(k)$	0.793	22.936	30.329	34.586	10.871	0.2

**Table 3.3b** The difference from the average, after Tab. 3.3a

$ \Delta H_R(k) $ radii	0	1	2	3	4	5	$\Delta H_R$
0	8.907	3.236	3.429	8.786	6.529	0.257	220.198
5	0.307	0.736	7.429	3.386	9.129	1.943	154.405
10	0.493	2.364	7.129	0.186	4.829	0.643	80.421
15	0.793	3.364	6.229	0.614	2.929	0.043	59.704
20	0.793	3.064	5.129	1.014	1.929	0.157	41.097
25	0.793	3.264	3.829	0.614	0.929	0.257	27.25
30	0.793	2.664	2.829	0.814	0.229	0.157	16.469
35	0.793	2.164	1.129	0.514	0.571	0.257	7.243
40	0.793	2.964	0.071	0.286	1.671	0.287	12.259
45	0.793	2.564	2.071	1.586	1.971	0.257	17.958
50	0.793	0.964	4.171	0.986	3.071	0.257	29.424
55	0.793	0.336	5.771	0.014	4.371	0.257	53.218
60	0.793	2.736	7.671	0.314	3.971	0.257	82.892
128	0.793	16.536	17.371	11.114	10.871	0.257	818.021

Table 3.4 Distribution of the directions after Top-Contouring, and Consistency Ratio Mapping (image in Fig. 3.1a)

segmentation	Directions					
	0 undeci- ded	1 horizon- tal	2 down- left	3 vertical	4 down- right	5 random
Top-Contouring	0	15129	33191	128861	61248	19636
Consistency Ratio	0	18255	34879	128021	62339	14571

Table 3.5 Comparison of Top-Contouring and Consistency Ratio Mapping pixel by pixel (Figs. 3.1d and 3.9)

		Consistency Ratio					
		0	1	2	3	4	5
T O P   C O N T O U R I N G	0	0	0	0	0	0	0
	1	0	12514	730	56	968	861
	2	0	815	27756	3422	2	1196
	3	0	29	3847	115349	7018	2618
	4	0	2837	66	5739	51086	1520
	5	0	2060	2480	3455	3265	8376

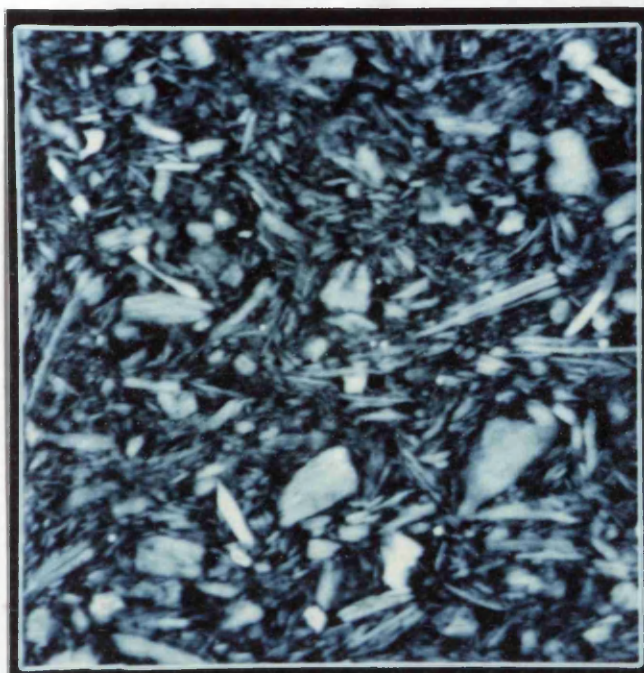


Fig. 3.1a Scanning electron micrograph of soil at 2,000 magnification picture width = 20  $\mu\text{m}$ .



Fig. 3.1b Differentiation of Fig 3.1a



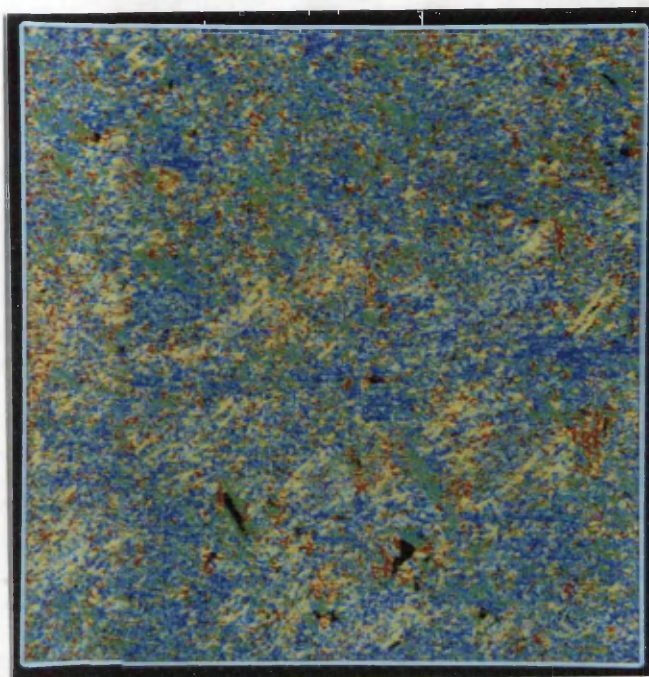


Fig. 3.1c Encoded image of Fig. 3.1b.

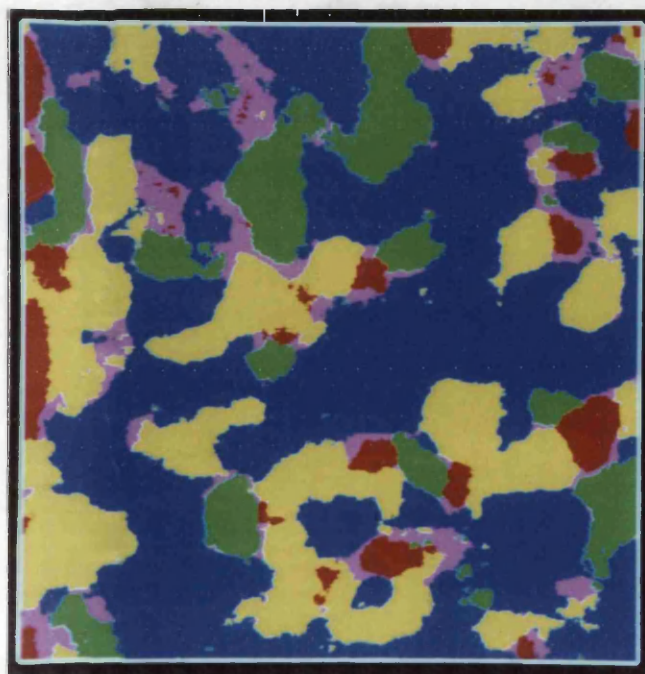


Fig. 3.1d Smoothed image of Fig. 3.1c

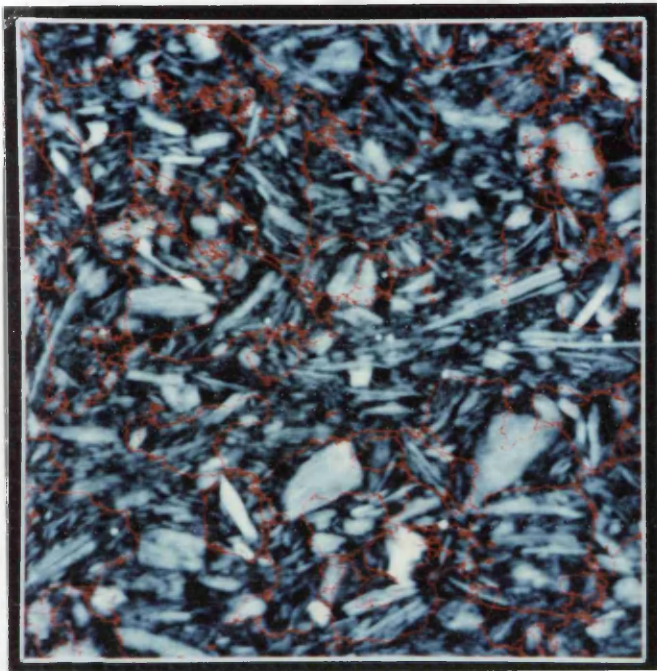


Fig. 3.1e Top-Contouring mapped image of Fig. 3.1a

			⋮		
	22	15	10	14	21
	16	6	2	5	13
	11	3	0	1	9
	17	7	4	8	20
	23	18	12	19	24
			⋮		

Fig. 3.2 Numbering of pixels within the mask

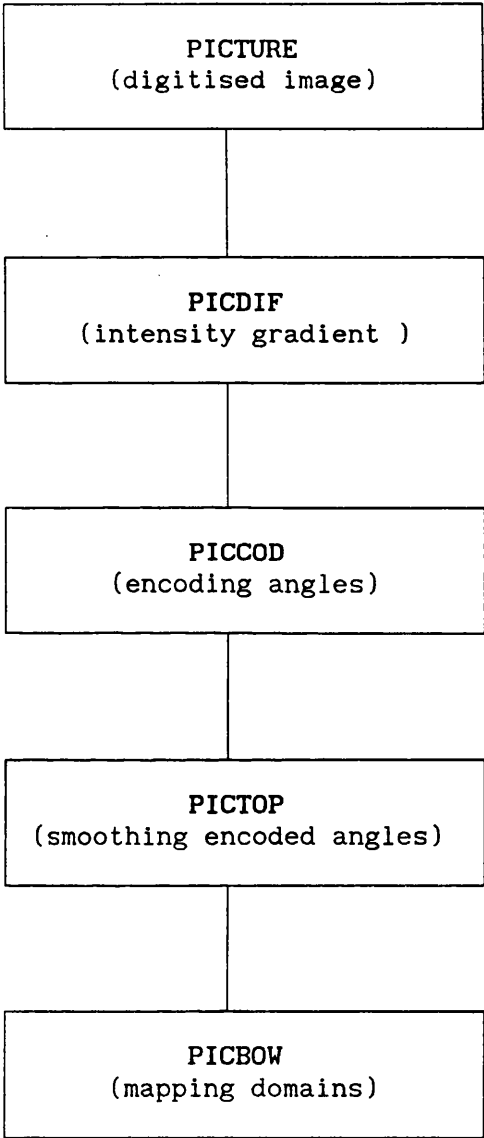


Fig. 3.3 The process of Top-Contouring.



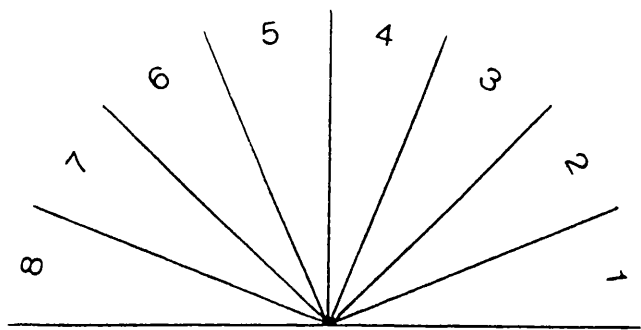


Fig. 3.4a Encoding angles to 8 directions within  $\pi$

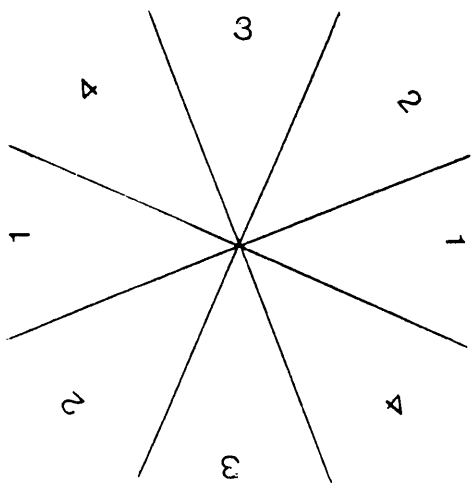


Fig. 3.4b Encoding angles to 4 direction within  $2\pi$

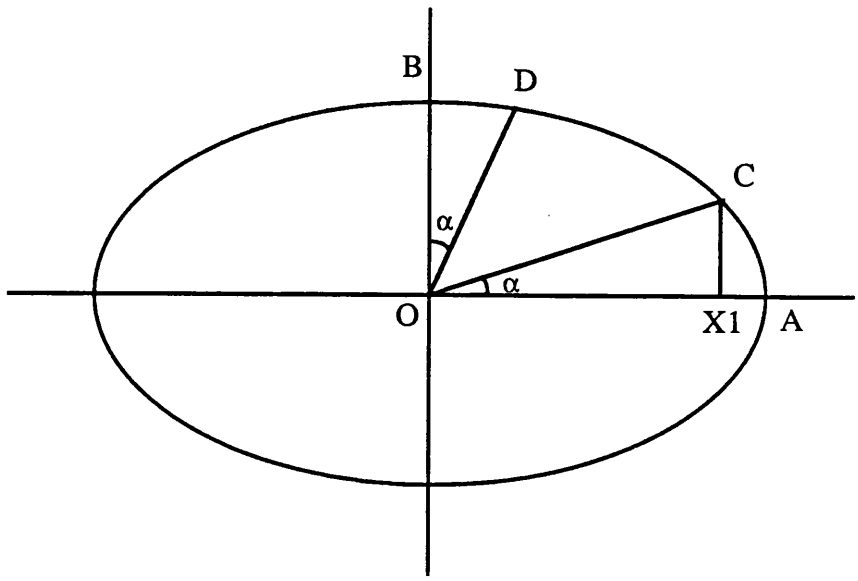


Fig. 3.5a Ellipse filter model for calculating  $EXC$ , with long axis ' $2a$ ' and short axis ' $2b$ ', area  $OAC = MAX/4$ ,  $OBD = MIN/4$ .

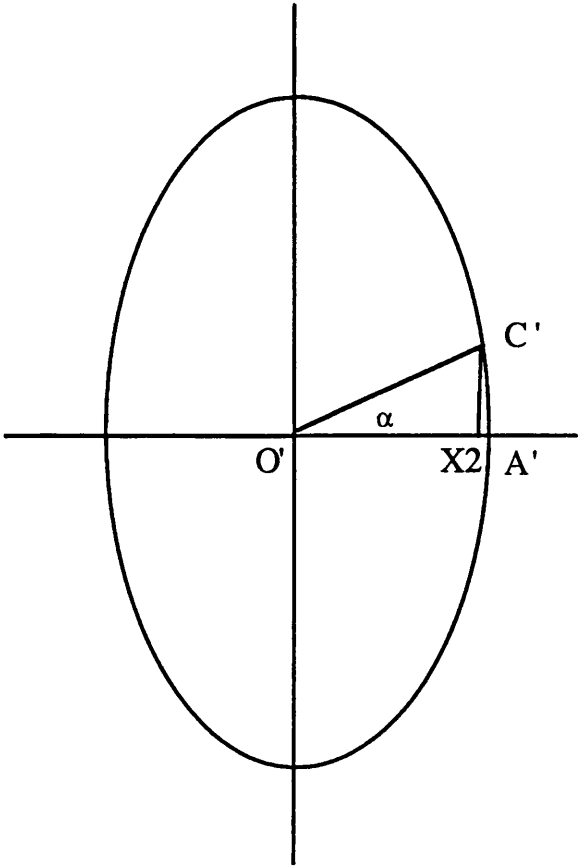


Fig. 3.5b Same ellipse filter model turned  $90^\circ$  degree. for easier calculation, here  $O'A'C' = OBD$ ,

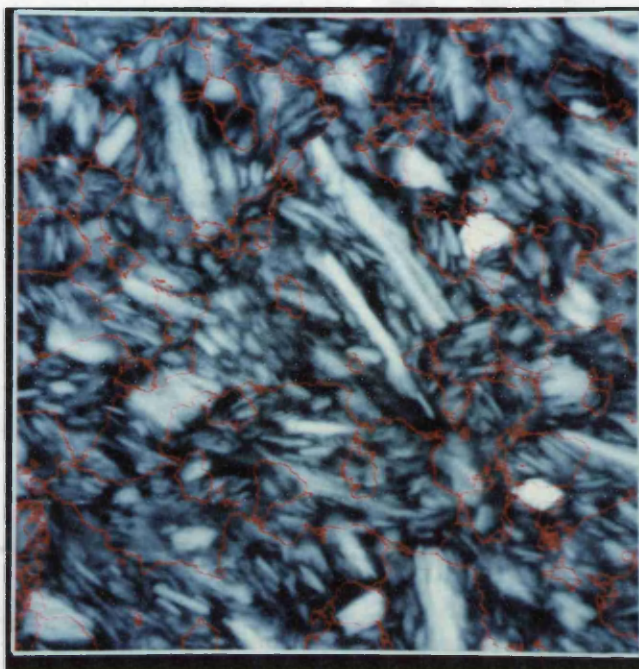


Fig. 3.6 Scanning electron micrograph with magnification 2,000. 20 pixels radius has been chosen for Top-Contouring Mapping.

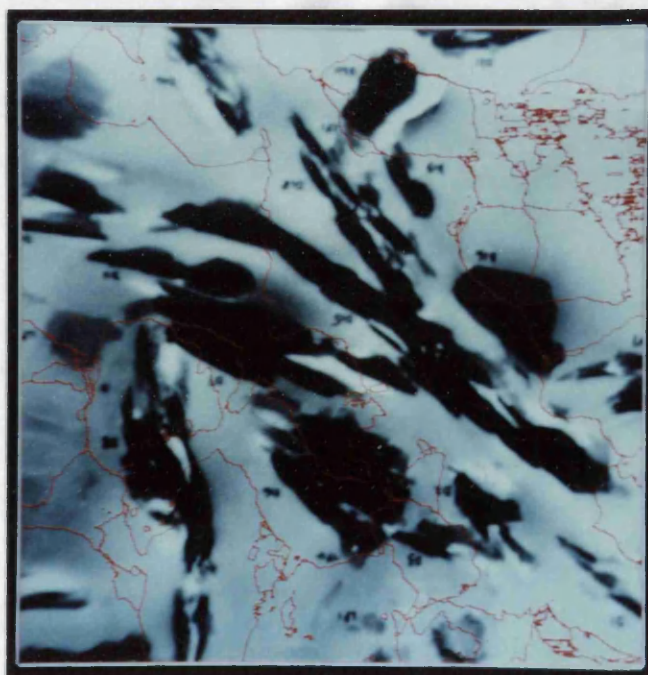


Fig. 3.7 Transmission electron micrograph with picture width 4  $\mu\text{m}$ . 35 pixels radius has been chosen for the Top-Contouring Mapping.

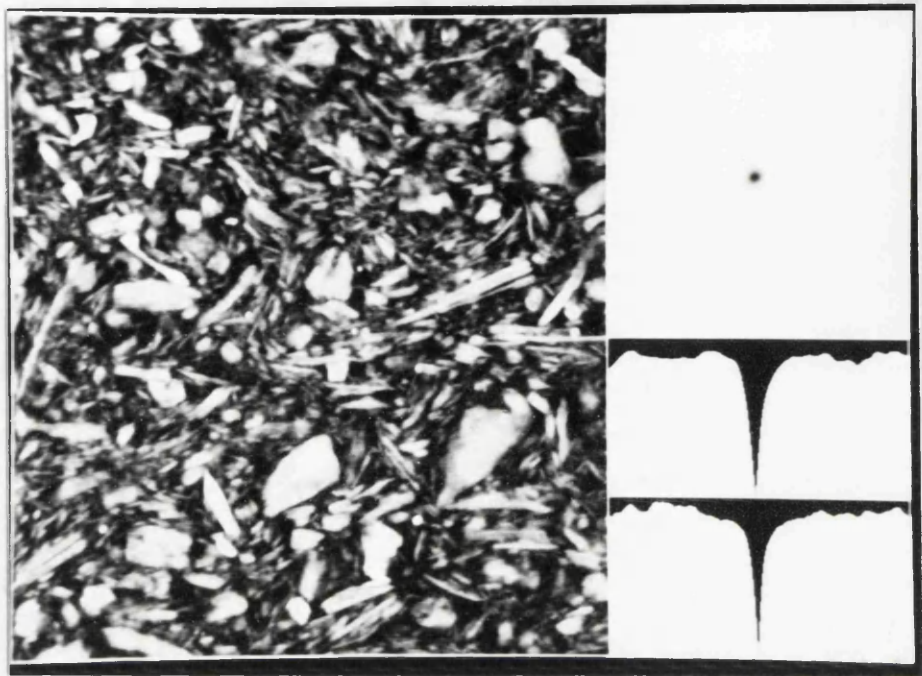


Fig. 3.8a Semi-variogram of Fig 3.1a. The original image is on the left. The Semi-variogram is at top right, the grey levels have been increased to improve visibility, the half-width of the semi-variogram is 128 pixel. Four <sup>half</sup> cross-sections of the semi-variogram are also shown, arranged in this picture:

horizontal                      vertical  
down-right                      down-left  
the linear scale of the diagonal pair is compressed.

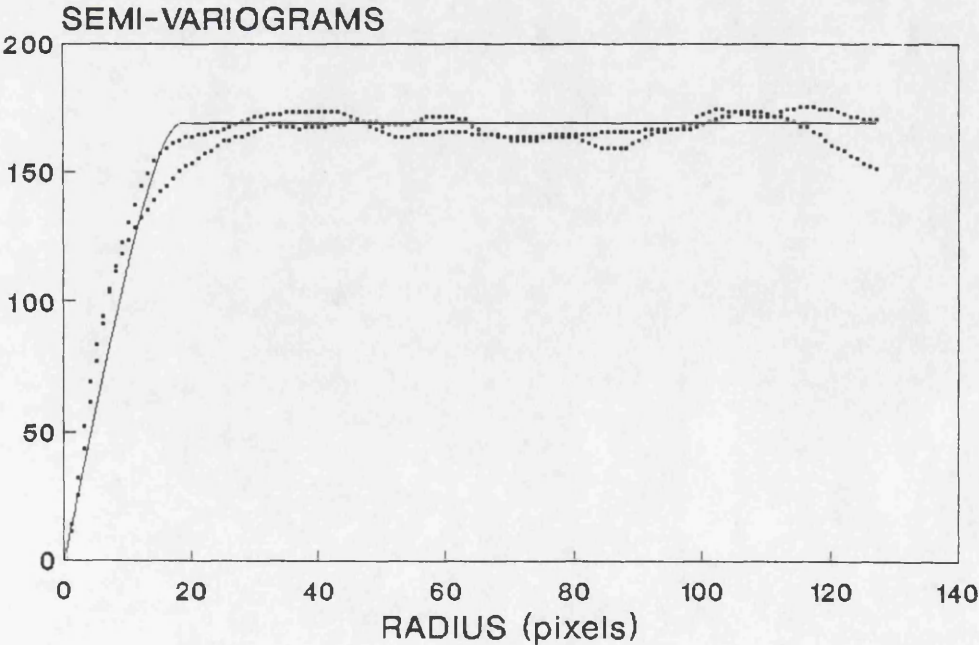


Fig. 3.8b Scaled up semi-variogram.



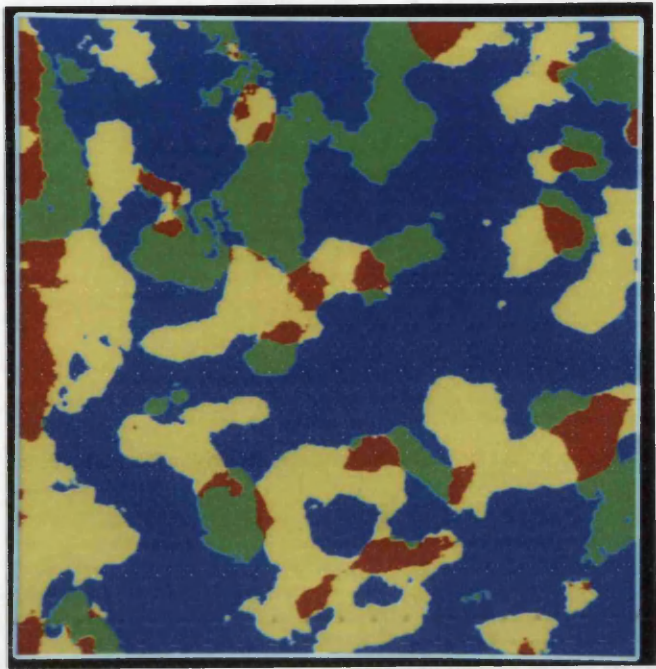


Fig. 3.9a summed intensity gradient vector of Fig. 3.1a by Consistency Ratio Mapping.

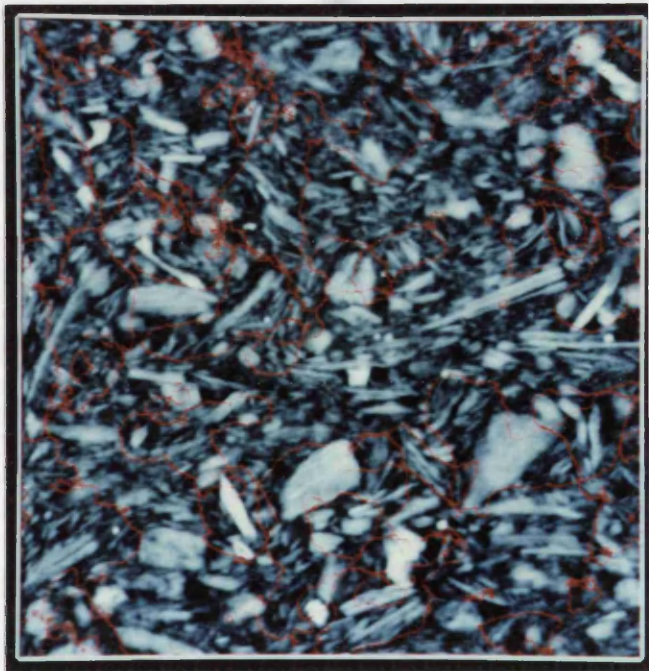


Fig. 3.9b Segmented Fig. 3.1a by Consistency Ratio Mapping.

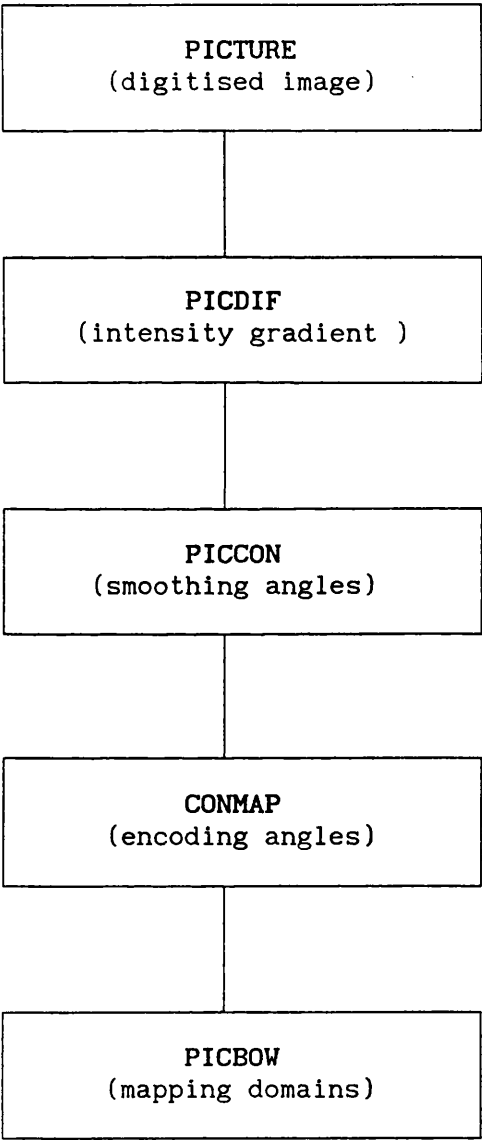


Fig. 3.10 The process of Consistency Ratio.

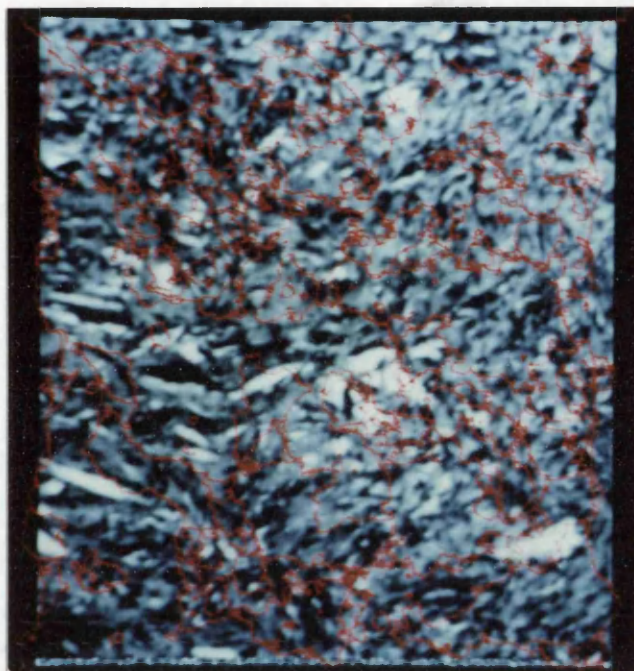


Fig. 3.11 Ultra-thin section of Kaolin on which the Top-Contouring mapping has subdivided horizontal and down-left domains from hand mapping random areas.

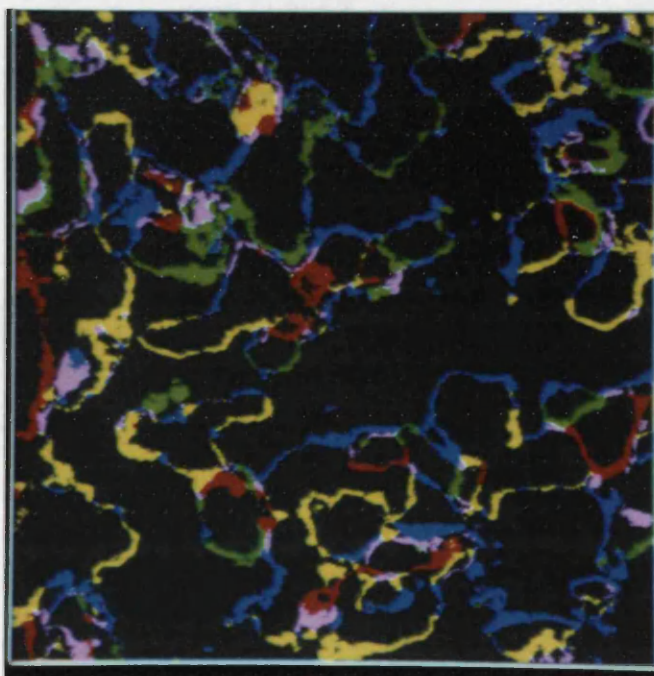


Fig. 3.12 Comparison of two segmentations, with similar areas in grey, and the Top-Contouring colours in areas which differ for Fig. 3.1a.



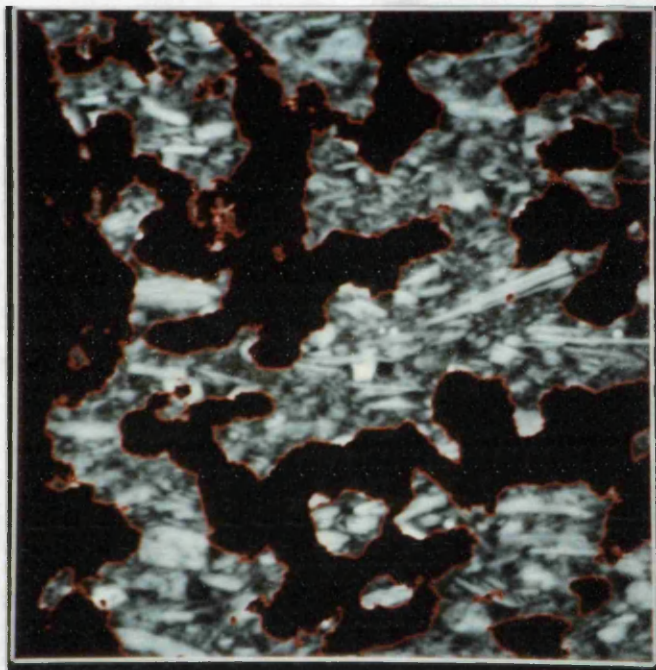


Fig. 3.13a Horizontal domains of image in Fig. 3.9 b.

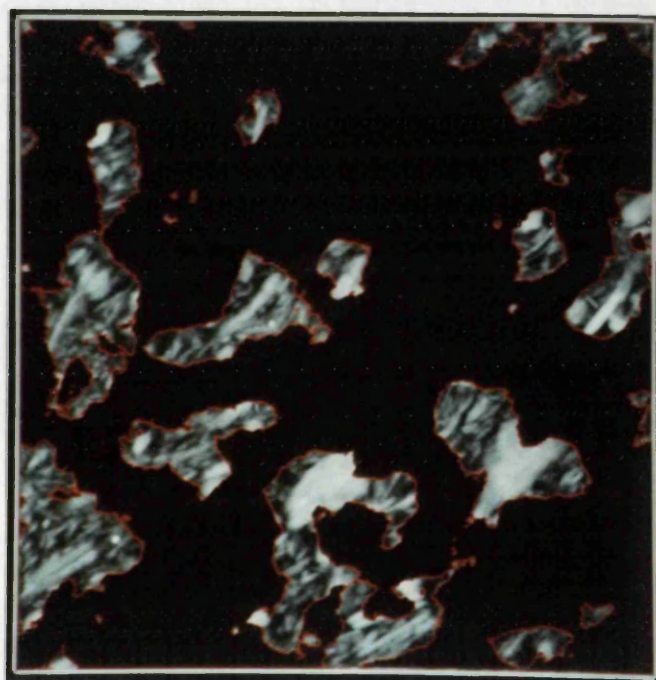


Fig. 3.13b Down-left domains of image in Fig. 3.9 b.



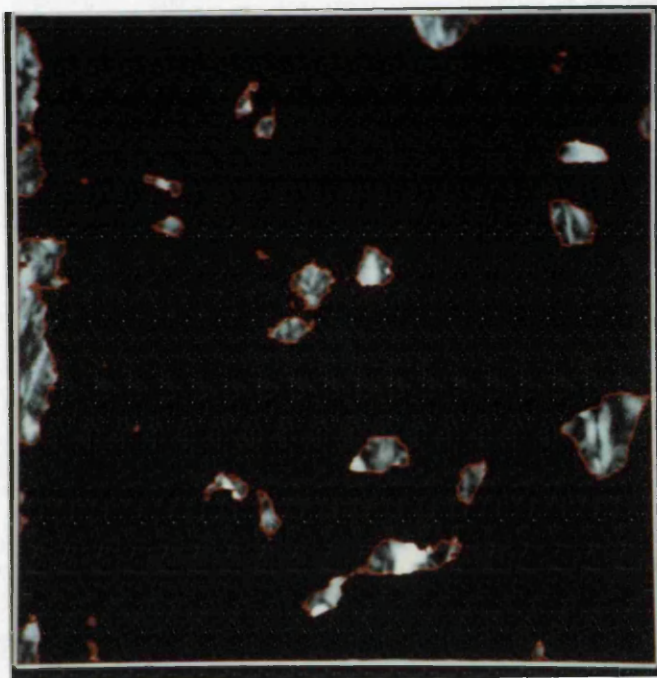


Fig. 3.13c Vertical domains of image in Fig. 3.9 b.

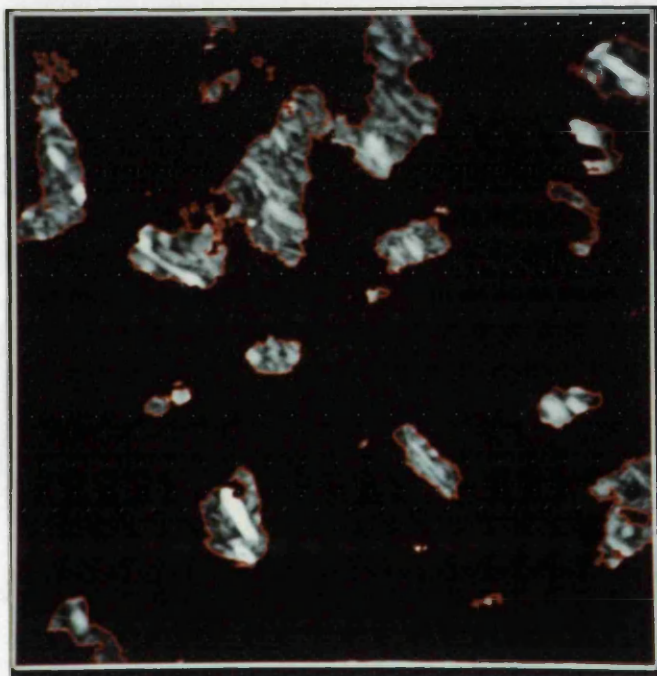


Fig. 3.13d Down-right domains of image in Fig. 3.9 b.

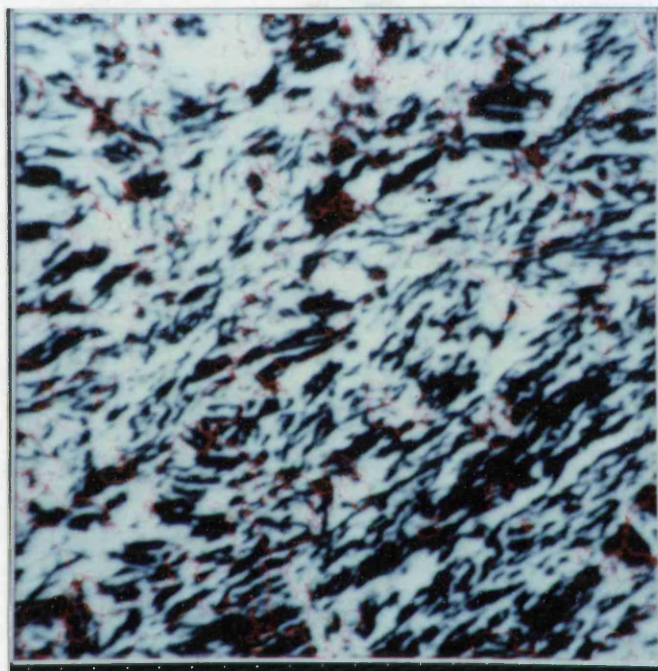


Fig. 3.14 Consistency Ratio Mapping on small scale image,  
(transmission electron micrograph from Smart 1966)

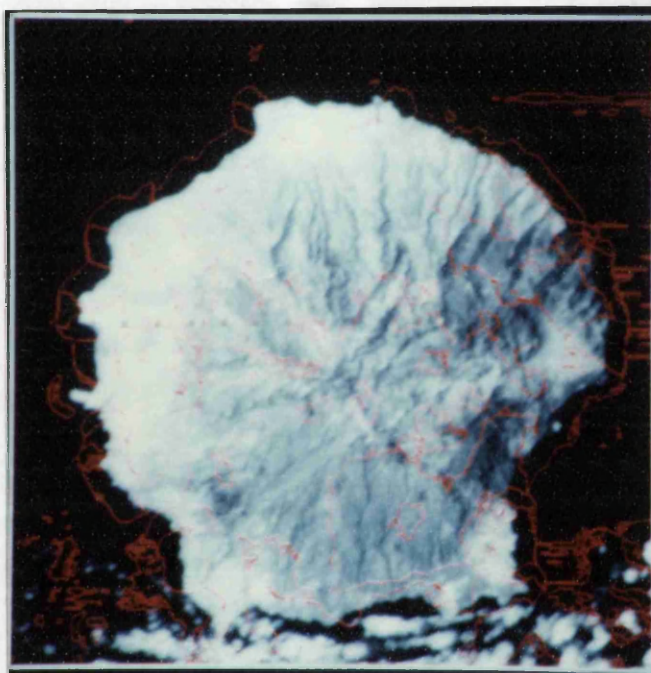


Fig. 3.15 Consistency Ratio mapping on air photo of an unidentified island.

## CHAPTER 4 GLOBAL ORIENTATION ANALYSIS

### 4.1 INTRODUCTION

In the last Chapter the author discussed two segmentations which could be used to analyse the oriented particles in groups (domains). In this Chapter a few new methods of studying the global orientation distribution of scanning electron micrographs will be presented.

The directional data statistics algorithm, which is presented first, is a synthesis of the theory of the intensity gradient method and the vector method of orientation analysis. In this synthesis, the modified mean direction and Consistency Ratio calculation have been used to enhance the intensity gradient, thereby eliminating the noise from the orientation distribution and revealing a more meaningful global histogram of direction. Local variations of porosity are also investigated by the method.

The Cooperation with King's College of London University on the global orientation analysis by using the Hough transform has produced a result which is comparable with that obtained by directional data statistics; and the cooperation with the Department of Electronics and Electrical Engineering of Glasgow University using the methods of Directed Vein and Convex Hull has also given good results.

## 4.2 METHODS OF DIRECTIONAL DATA STATISTICS

### 4.2.1 Effects of Intensity Gradient

Unitt (1975) was the first to use the polar histogram of the intensity gradient vector angle of the image to indicate the global orientation distribution of the features within the micrograph (detail of intensity gradient calculation see Section 3.2). Fig. 4.1 is an example of Unitt's polar histogram.

Subsequently, Unitt (1976) replaced Equation 3.4a and 3.4b by a more accurate approximation obtained by applying the linear form of Taylor's expansion to 5 collinear points; and Tovey and Smart (1986) suggested leaving  $\alpha$  'undecided' when  $U$  is less than a threshold value. To improve the accuracy and reduce the spikes of the polar histogram in Fig. 4.1, firstly the author used the 20,14-formula introduced by Smart and Tovey (1988) to calculate the intensity gradient vector (see Section 3.2). This was done because the 20,14-formula uses a more nearly circular mask, and it smoothes out the variance in the mask. Secondly, in the author's code, for  $U < 3$ , the angle of intensity gradient  $\alpha$  was labelled 'undecided' and omitted from the histogram (detail see Section 3.3.2). Thirdly, when plotting the polar histogram, a  $5^\circ$  interval has been used. Fourthly, the digitisation used now is 256 grey levels, whereas Unitt may have had to use a coarser subdivision of the range. All these reduce the spikes of the histogram and produce a smoother 'ellipse'. Fig. 4.2 is a typical scanning electron micrograph of clay. Fig. 4.3a and Fig. 4.3b are the polar histogram of unweighted and weighted intensity gradient vector of Fig. 4.2 calculated by using the 20,14-formula. These two



histograms give crude measures of the preferred orientation of the whole image.

Several factors, which suggest that the histogram of angle should perhaps be weighted by the strength of the intensity gradient, will be discussed below.

Fig. 4.4 gives an example. If the upwards facing edges of a particle are sharp and the downwards edges are diffuse, see Fig. 4.4c, there will be more downwards pixels contributing to the histogram. Therefore the histogram of angles will be asymmetrical, see Fig. 4.4d. However, in the example given here, the deviations from symmetry in both the unweighted and weighted histograms, Figs. 4.3a and 4.3b, are negligible.

Another case arises if all the edges at the particle's ends are not as sharp as the side edges, then the unweighted polar histogram of angles will be distorted. So weighting the angles,  $\alpha$ , by the strength of the intensity gradient,  $U$ , will give a clearer result. It is evident from Figs. 4.3a and 4.3b that the histogram weighted by strength is thinner than the unweighted histogram. Therefore, the average strength in the direction of the shorter axis is slightly smaller than in the direction of the longer axis.

However, on the other hand, it has been noticed that there are some particles brighter than others in the images, because they have been charged up or are composed of a different material. In this case, the weighted histogram may give too much prominence to the brighter particles (which have higher values of  $U$  at their edges), so the unweighted histogram may be better.

#### 4.2.2 Enhanced Orientation Information

Fig. 4.4a shows a single particle and Fig. 4.4b is the corresponding polar histogram. In addition to the correct maxima from the faces, there are small spurious maxima from the ends. This may be one reason that the unweighted and weighted histograms of angle have elliptical shapes. In order to eliminate the effect of the particles' ends and of the particles lying in the less popular direction, it was therefore decided to smooth the intensity gradient image to obtain synthetic and enhanced orientation information.

Normally filtering operations for smoothing are based on the original grey level images. Because the intensity gradient image is a vector image, the filtering operation has to be based on the directional statistics theory. Therefore, in the present work the summed vector of the observed intensity gradient vectors within a large circular filter has been used to enhance the direction information (Mardia 1972, namely mean direction). The details of calculation are in Section 3.4.2. To match the texture of the image, a large circular filter was employed using the mechanism described in Section 2.2.6 to obtain the mean vector. The calculation was carried up to the edges of the secondary image. The filter size was 20 pixels radius (Section 3.3.3).

The resultant of the summed vector within the filter was used to plot the unweighted and weighted polar histograms as shown in Fig. 4.3c and Fig. 4.3d. Both Fig 4.3c and 4.3d measure the amount of preferred orientation. Since the figures are the enhanced results,

they show stronger and more obvious distributions than the histograms of intensity gradient with bow tie shape. The less frequent directions, such as those around the short axes of the ellipses in Figs. 4.3a and 4.3b, have been smoothed away. Because Figs. 4.3c and 4.3d are based on the summed vector within the large filter, they express the information of the group of particles within the filter, and eliminate the effect of the few particles which were not in the most popular direction and of the particles' ends. These two figures indicate that at this scale of filtering most of the particles in the image were in the direction between  $40^\circ$  and  $60^\circ$ . Within the main tendency, there were two sub-peaks at  $40^\circ$  and  $60^\circ$  with even stronger local preferred orientation. Thus Fig. 4.3c and 4.3d give more detailed directional information than did Figs. 4.3a and 4.3b.

In Fig. 4.2, it can be seen that the micrograph can be divided approximately into two parts, the left side has most particles lying in the direction of  $40^\circ$ ; and the right side has most particles lying in the direction of  $60^\circ$ . Comparison of Figs. 4.3c and 4.3d shows that the  $40^\circ$  sub-peak has a stronger local preferred orientation than the  $60^\circ$  sub-peak. *i.e.* in general speaking particles in left side are longer and thinner than those in right side and with better parallelism. On the other hand the area of  $60^\circ$  sub-peak is bigger than that of  $40^\circ$  sub-peak. which may suggest that there is less order within areas mapped to  $60^\circ$  than areas mapped to  $40^\circ$ .



### 4.2.3 Average Summed Vector and Consistency Ratio

If and only if the intensity gradient vectors within the filter have strong strength and point mostly in the same direction, the mean direction vector can be large. The average strength of the mean direction (summed vectors) over the whole image have been calculated to give the perfection of local preferred orientation and to provide a further method of showing whether there are some directions in the micrograph having particles lying closely with each other parallelly.

Fig. 4.3e is a polar histogram of the average strength of the mean direction vector (i.e. of the value of Fig. 4.3d divided by the value of Fig. 4.3c) which gives a measurement of the perfection of preferred orientation in each direction. Fig. 4.3e again shows a maximum in the direction of  $40^\circ$ , with a sub-peak over the main tendency. This indicated that in the direction of  $40^\circ$  of the micrograph, not only does a large population of particles lie in this direction, they are of long and thin shape, but also these particles are packed together very closely, i.e. with good parallelism.

As an alternative to the average mean direction, the consistency ratio can also be used to analyse orientation preferences of the particles in the image (see Section 3.4). The average consistency ratio within each  $5^\circ$  interval is plotted in Fig. 4.3f. This figure again measures the perfection of local preferred orientation in each direction and shows stronger parallelism in the inclined direction of  $40^\circ$ . Although there are some differences between Figs. 4.3e and 4.3f, they both give the packing characteristics of particles against orientation on a pixel-by-pixel basis. In some ways, the method in

Fig. 4.3f may be more realistic than Fig. 4.3e, because Fig. 4.3f corrects both for variations in the number of edges per unit area and also for variations in the range of intensity. On the other hand, Fig. 4.3e may be better at pointing to features of interest.

In summary, Fig. 4.3a and 4.3b give crude measurements of the preferred orientation of the whole image. Fig. 4.3c and 4.3d are the measurements of preferred orientation on a domain-by-domain basis after the noise and rogue particles have been smoothed away. Fig. 4.3e and 4.3f show the perfection of the local preferred orientation.

For comparison with the textural analysis methods described in Chapter 3 and Chapter 5, the calculations are also reported for the image of Fig. 3.1a. Fig. 4.5 gives the histograms of unweighted and weighted intensity gradient and of unweighted and weighted mean direction over the image of Fig. 3.1a. These figures give the preferred orientation between  $85^\circ$  and  $110^\circ$ , and have sub-peaks at  $90^\circ$  and  $105^\circ$ .

#### 4.2.4 Local Void Ratio

Variations of local porosity,  $\rho$ , are significant both for classification of the microstructure and interpretation of results from experiments. In particular the local porosity may vary systematically with the direction of local preferred orientation. This is a factor, which, if it is significant, will have to be taken into account in micromechanical computations.

In order to give a large field of view, some scanning electron

micrographs have been taken at too small a magnification to produce a black and white picture. The histogram of grey levels in the original image lacks two distinct peaks with a distinct and low trough between them; so grey-level discrimination fails to distinguish particles from voids. The modified method of using local histograms for parts of the image (Gonzalez and Wintz, 1987, p362) also failed. An approximate analysis was therefore made using:

$$\rho = \frac{I_s - I}{I_s - I_v} \quad (4.4)$$

where  $I_s$ : intensity for solid particle,  
 $I_v$ : intensity for void,  
 $I$ : intensity at the pixel with certain local mean direction.

In effect, this is a system of electron beam micro-analysis analogous to X-ray micro-analysis. The principal difficulty in using eqn 4.4 lies in choosing  $I_s$  and  $I_v$ . If possible,  $I_v$  should be measured from a block of pure resin similar to that used to fill the voids; this measurement should be made at the same time and under the same conditions as the main microscopy.  $I_s$  is more difficult, perhaps the best method would be to take measurements from both quartz and alumina standards, and to compute from these the value for the particles in the sample, which are alumina-silicates. In practice, approximate value of  $I_s$  and  $I_v$  had to be chosen using the histogram of intensity as a guide. The advantage of the grey scale method of calculating porosity is that it is not limited by resolution. It is at present thought that the orientation analysis can be used for smaller magnification images than have been used so far, which would cover larger areas; and the grey scale method of calculating porosity would

also be suitable for these images.

The polar histogram of local porosity against direction is shown in Fig. 4.6. As a basis of comparison, the figure also shows the average porosity calculated from the grey levels for the whole image (this is not to be confused with the average in each direction). Comparison of Fig. 4.4 with Figs. 4.3e and 4.3f shows that there is a tendency for the porosity to be less in the directions with stronger local preferred orientation, *i.e.* where the particles are more closely packed together parallelly.

#### 4.2.5 Enhanced Orientation Mapping

Further work which could be done following the directional data statistics is Enhanced Orientation Mapping. This arises by analysing the polar histograms of either the unweighted or weighted mean direction vector (Figs. 4.3c and 4.3d). The histograms have two sub-peaks in the main tendency, which show that in these two directions,  $40^\circ$  and  $60^\circ$ , particles lie very closely parallel with each other and form very tightly packed domains. Thus, these two directions could be used to segment very tightly packed domains, rather than using the specified directions for segmentation as described in Chapter 3. These domains will be very important microstructures in measuring soil mechanic characteristics. This new segmentation based on the result of Enhanced Orientation Analysis is called Enhanced Orientation Mapping.

### 4.3 HOUGH TRANSFORM TECHNIQUE

The Hough transform is a pattern recognition technique which has been widely used recently. However, the detected objects, to which the Hough transform is applied, are normally very large, and mostly there have been only a few or even one feature in a single image, and the features have had very simple outlines. The present work is thought to be a new effort to use the Hough transform to analyse very complicated and very irregular images like scanning electron micrographs of clay, such as Fig. 4.7a, which have hundreds of particles in them.

Clay particles normally have thin long shapes, so the edges of the particles can be regarded as lines for employing the Hough transform to analyse the orientation of the particles.

#### 4.3.1 Theory of Hough Transform

Hough (1962) proposed a method to analyse the specific structural relationships between pixels in the image. This method has been used in the cooperation with King's College of London University to analyse the edge structures of particles in soil micrographs.

Consider a point  $(x_1, y_1)$  in the image. The general equation of a straight line through this point in slope-intercept form gives:

$$y_1 = ax_1 + b. \quad (4.5)$$

There is an infinite number of lines that pass through the point  $(x_1, y_1)$ , but they all satisfy the equation  $y_1 = ax_1 + b$  for varying

values of  $a$  and  $b$ . However, if we write this equation as  $b = -x_1 a + y_1$ , and consider the  $a$ - $b$  plane (normally called Hough slope-intercept space), then we have the equation of a *simple* line for a fixed pair  $(x_1, y_1)$ . Furthermore, a second point  $(x_j, y_j)$  in the image will also have a line in Hough space associated with it, and this line will intersect the line associated with  $(x_1, y_1)$  at  $(a', b')$ , where  $a'$  is the slope and  $b'$  the intercept of the line containing both  $(x_1, y_1)$  and  $(x_j, y_j)$  in  $x$ - $y$  space. These concepts are illustrated in Fig. 4.8. In fact, every point lying on a line in  $x$ - $y$  space will have lines in Hough space that intersect at  $(a', b')$ . Thus in the simple example given, the line in  $x$ - $y$  space can be found by searching Hough space for the point  $(a', b')$  through which more than one line passes. In general, several lines in  $x$ - $y$  space can be found by searching Hough space for its maxima. In practice, Hough algorithms are implemented by replacing continuous Hough space by a discrete two-dimensional histogram, which is usually called an accumulator array.

A problem with using the equation  $y = ax + b$  to represent a line is that both the slope and intercept approach infinity as the line approaches a vertical position. One way around this difficulty is to use the normal representation of a line, given by:

$$x \cos\theta + y \sin\theta = \rho \quad (4.6)$$

The meaning of the parameters used here is illustrated in Fig. 4.9a. Instead of straight lines, however, we now have sinusoids as loci in the  $\rho$ - $\theta$  Hough space. As before,  $M$  collinear points lying on a line  $x \cos\theta_j + y \sin\theta_j = \rho_1$  will yield  $M$  sinusoidal curves that intersect at  $(\rho_1, \theta_j)$  in the Hough space. The range of angle  $\theta$  is  $\pm 90^\circ$ , measured

with respect to the  $X$  axis. Thus, with reference to Fig. 4.9b, a horizontal line has  $\theta = 0^\circ$  with  $\rho$  being equal to the positive  $x$  intercept. Similarly, a vertical line has  $\theta = 90^\circ$ , with  $\rho$  being equal to the positive  $y$  intercept, or ( $\theta = -90^\circ$ , with  $\rho$  being equal to the negative intercept). For the points 1,2,3,4,5 in Fig. 4.9b, there are five lines in Hough  $\rho$ - $\theta$  parameter space as shown in Fig. 4.9c. These lines intersect at point  $A$  (Fig. 4.9c) for line 1-3-5 (Fig. 4.9b), at  $B$  for line 2-3-4, and at  $C,D,E,F$  for line 2-5, 1-4, 1-2, 4-5 respectively.

#### 4.3.2 Hough Transform Application

The slope-intercept Hough space was employed to analyse the global orientation distribution of scanning electron micrographs of clay (Costa *et al* 1991a, 1991b). The calculations were suggested by the author and made by Costa. The original micrograph, Fig. 4.7a, was firstly processed by edge detection as shown in Fig. 4.7b; then a Hough transform was used to recognize the straight lines of edges; thence the global orientation distribution of particles was expressed by summing up the angles of these straight lines and plotting them on a polar histogram.

The implementation was based on Eqn 4.7. For easy calculation, *i.e.* to keep the slope of the line,  $a$ , smaller than 1 and larger than 0, the whole slope-intercept space ( $0^\circ - 180^\circ$ ) was divided into four accumulator arrays, where  $AB_1 = 0^\circ - 45^\circ$ ,  $AB_2 = 135^\circ - 180^\circ$ ,  $AB_3 = 45^\circ - 90^\circ$ , and  $AB_4 = 90^\circ - 135^\circ$ , each with dimension  $N_a \times (N_b + 1)$ , where the image size is  $(N+1) \times (N+1)$ . Then taking  $\Delta A = 1/N_a$ ,  $\Delta B = 2N/N_b$ , for every edge element  $(x,y)$  in the image, do:

For  $a_1 = 0 \text{ --- } N_a - 1$

$$b_1 = (y - xa_1\Delta A + N)/\Delta B$$

$$AB_1(a_1, b_1) = AB_1(a_1, b_1) + 1 \quad (4.7a)$$

For  $a_2 = 1 \text{ --- } N_a$

$$b_2 = (y + xa_2\Delta A)/\Delta B$$

$$AB_2(a_2, b_2) = AB_2(a_2, b_2) + 1 \quad (4.7b)$$

For  $a_3 = 1 \text{ --- } N_a$

$$b_3 = (x - ya_3\Delta A + N)/\Delta B$$

$$AB_3(a_3, b_3) = AB_3(a_3, b_3) + 1 \quad (4.7c)$$

For  $a_4 = 0 \text{ --- } N_a - 1$

$$b_4 = (x + ya_4\Delta A)/\Delta B$$

$$AB_4(a_4, b_4) = AB_4(a_4, b_4) + 1 \quad (4.7d)$$

The slope ( $A$ ) and intercept ( $B$ ) of confirmed straight lines can be obtained from the corresponding peak coordinates,  $(a_k, b_k)$ , by using the appropriate expression below (e.g. if the peak is in  $AB_k$ , use equ. 4.8 a and 4.8 b and so on).

$$A = a_k \Delta A \quad (4.8a)$$

$$B = b_1 \Delta B - N \quad (4.8b)$$

$$A = -a_k \Delta A \quad (4.8c)$$

$$B = b_2 \Delta B \quad (4.8d)$$

$$A = 1/(a_k \Delta A) \quad (4.8e)$$

$$B = -Ab_3 \Delta B - N \quad (4.8f)$$

$$A = -1/(a_k \Delta A) \quad (4.8g)$$

$$B = -Ab_4 \Delta B \quad (4.8h)$$

In the present Hough transform implementation, due to the high



amount of noise in the edge detected image, the image has been segmented in squares of equal sizes,  $N_{sq}$ , in order to reduce the amount of false detected lines and the overall execution time. As each of these segments is treated separately (independent accumulator arrays), line segments with length larger than  $N_{sq}$  pixels may contribute with more than one vote to the histogram, which is in fact a desirable operation: the votes in the histogram will be linearly weighted by the length of the respective line segments.

In the analysis, the Hough transform has been performed with the following parameters:

$$N_a = 64$$

$$N_b = 64$$

$$N_{sq} = 32$$

maximum gap in the line segments = 1 pixel

minimum line length = 8 pixels

threshold for peak detection (accumulator array) = 8

Fig. 4.7c depicts the detected line segments after using the Hough transform, and the respective histogram for the orientations of the detected line segments is given in Fig. 4.10.

From Fig. 4.7c, it can be seen that the straight lines in the image have been detected, and that the background noise has been eliminated. The histograms for the orientations of the lines were obtained by updating the histogram position indexed by the slope of each detected line. The histogram in Fig. 4.10 shows the orientation of the particles focussing attention on the two major directions of preferred orientation and a third minor one, which produce much the

same pattern as shown in Fig 4.3.

As a comparison, some segmented images, which only contain the domains with preferred orientation in a certain direction, have also been analysed by the Hough transform. These segmented images were obtained by the methods of Chapter 3. Fig. 4.11a is an image with segmented down-left domains, Fig. 4.11b is the image with detected edges, and Fig. 4.11c is the result after the Hough transform. Fig. 4.11d is the polar histogram of the detected straight line of Fig. 4.11c, which gives an orientation exactly the same as the direction of the segmentation *i.e.* around  $45^\circ$ .

#### 4.4 DIRECTED VEIN AND CONVEX HULL

In addition to the methods reviewed above, the cooperation with the Department of Electronics and Electrical Engineering has produced two other methods to analyse the orientation of the particles within the clay micrograph. They are The Directed Vein and Convex Hull methods. Both of them are based on processing the chain code of the particles' boundaries. Both methods were developed and implemented by Lou (Lou et al 1991)

The Directed Vein method smooths the similarly oriented chain code, *i.e.* vein, then finds the turning points of the straightened veins. The linking line between two turning points at two ends of the straightened veins is obtained as the centre axis of the particle. The direction of the axis is taken as the orientation of the particle.

The Convex Hull method finds the enveloping polygon of a particle

as shown in Fig. 4.12. The direction of the longest diagonal of this polygon is taken as the orientation of the particle.

Fig. 4.13a shows an example of applying the Directed Vein to a scanning electron micrograph of clay. The particles are shown in outline, and the directed veins are shown by straight lines. Most orientations appear to have been correctly found. Fig. 4.13b is a polar histogram of the directed veins, which gives a global orientation distribution comparable with those given by the methods described in the previous two sections. The results from the Convex Hull method were also broadly similar.

#### 4.5 CONCLUSIONS

Unitt introduced the polar histogram of intensity gradient angles to indicate the global orientation distribution of a micrograph. A comparison of the weighted and unweighted polar histograms based on an improved intensity gradient calculation has been made here and suggests that the weighted polar histogram may give a more distinct directional distribution. By using the directional data statistics method, the orientation information has been enhanced, thereby giving a more obvious polar histogram and throwing attention on to sub-parallel domains. These sub-parallel domains could be segmented following the directions of the main sub-peaks obtained by the Enhanced Orientation Analysis; this segmentation is called Enhanced Orientation Mapping. Local porosity analysis following these methods showed the relationship between the local porosity and the local orientation of the particles. The other methods of analysing global orientation distribution include Hough transform, Directed Vein, and

---

Convex Hull, all of which gave polar histograms which were comparable to those obtained from the intensity gradient methods. In order to get better results from these methods, the author suggests using skeletons of particles (see Chapter 5) instead of edges in the methods of Hough transform and Directed Vein.

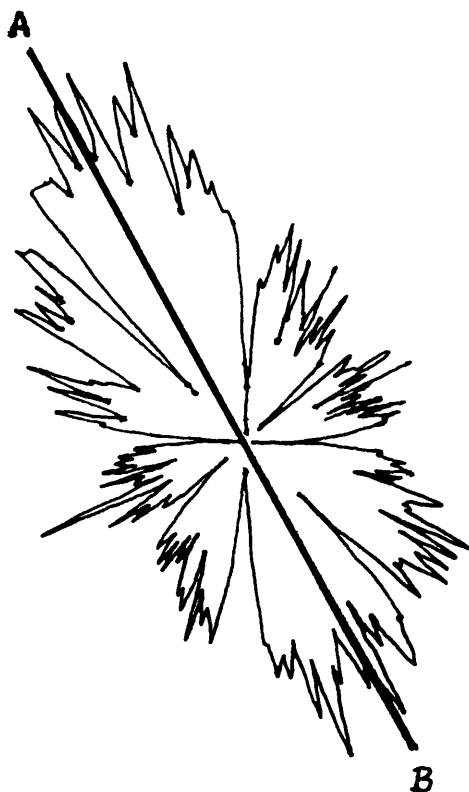


Fig. 4.1 Unitt's polar histogram shows the global orientation distribution of a scanning electron micrograph of clay (Unitt 1976).

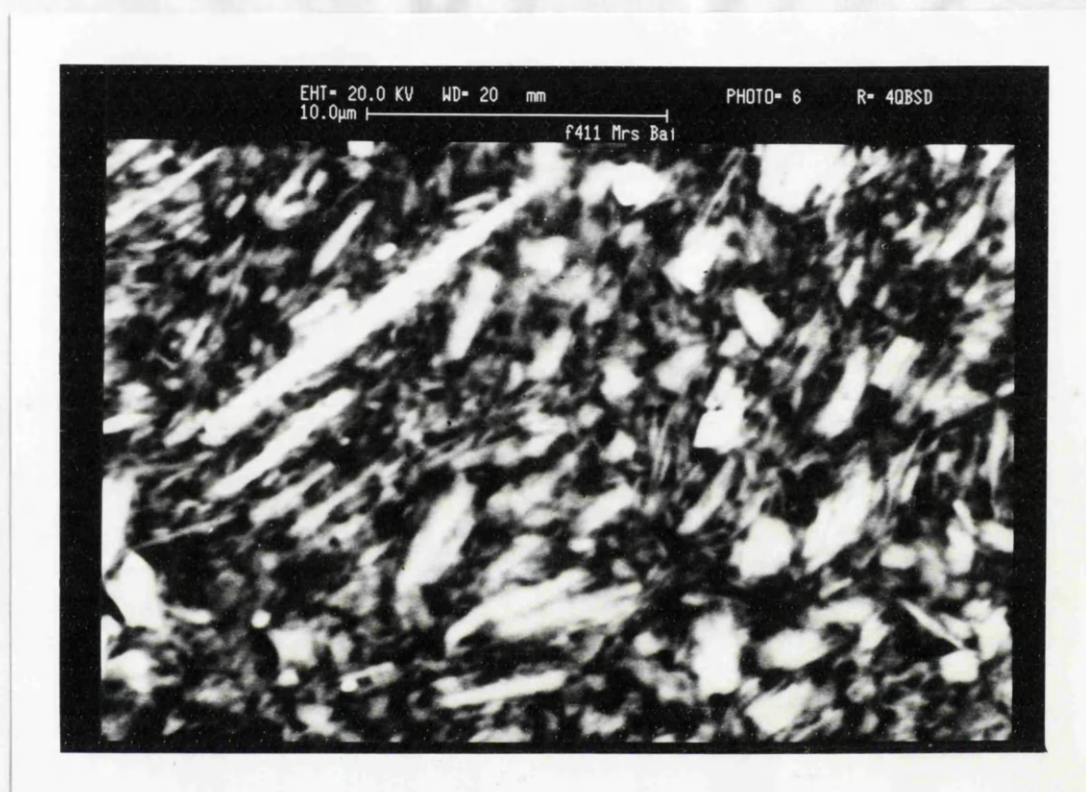
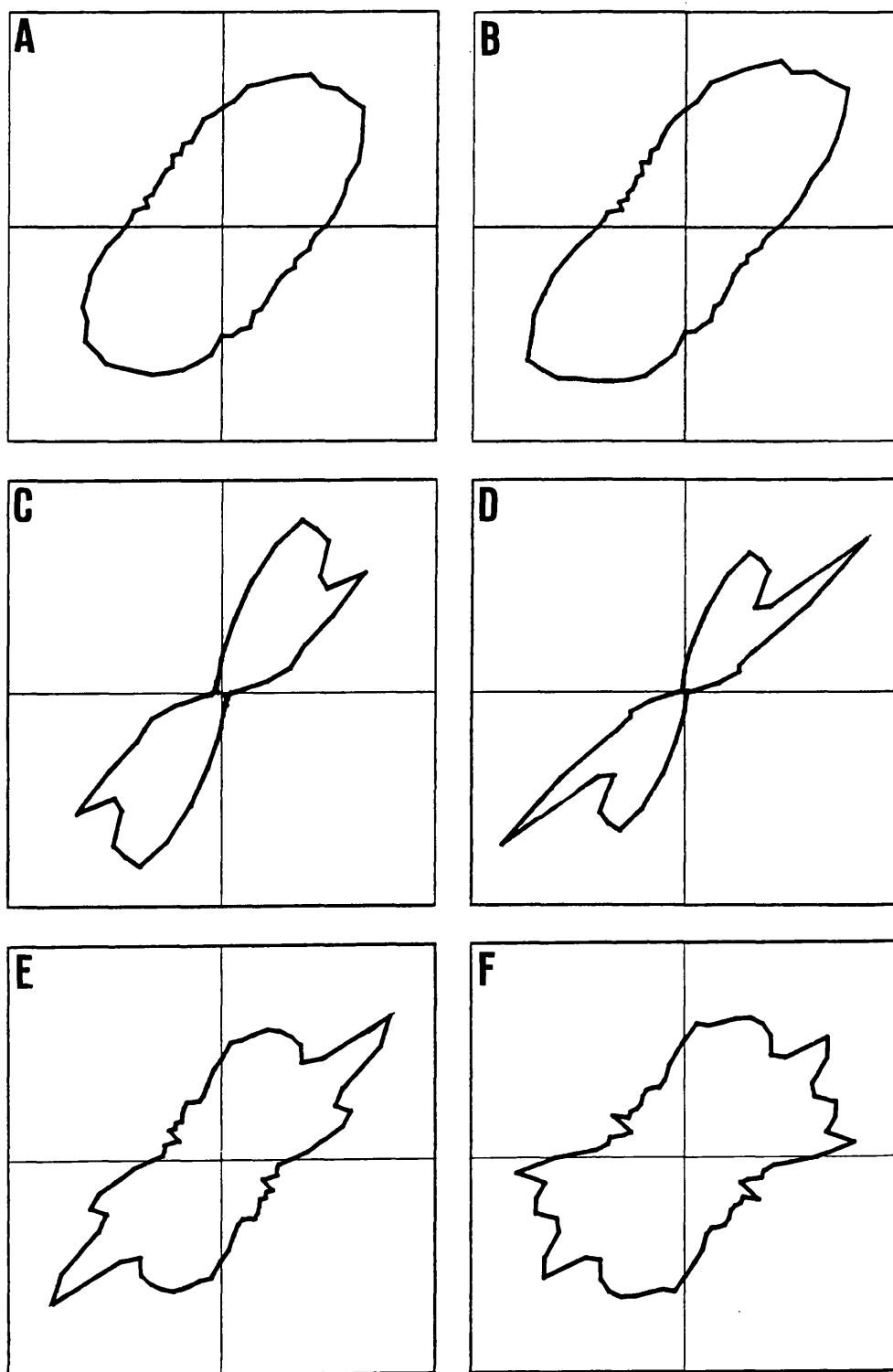


Fig. 4.2 Backscattered scanning electron micrograph of sheared kaolin



**Fig. 4.3** Polar histograms calculated for Fig. 3.1a  
a) unweighted histogram of angle of intensity gradient,  
b) weighted version of (a),  
c) unweighted histogram of mean direction,  
d) weighted version of (c),  
e) Average magnitude of mean direction vector,  
f) Histogram of Consistency Ratio.

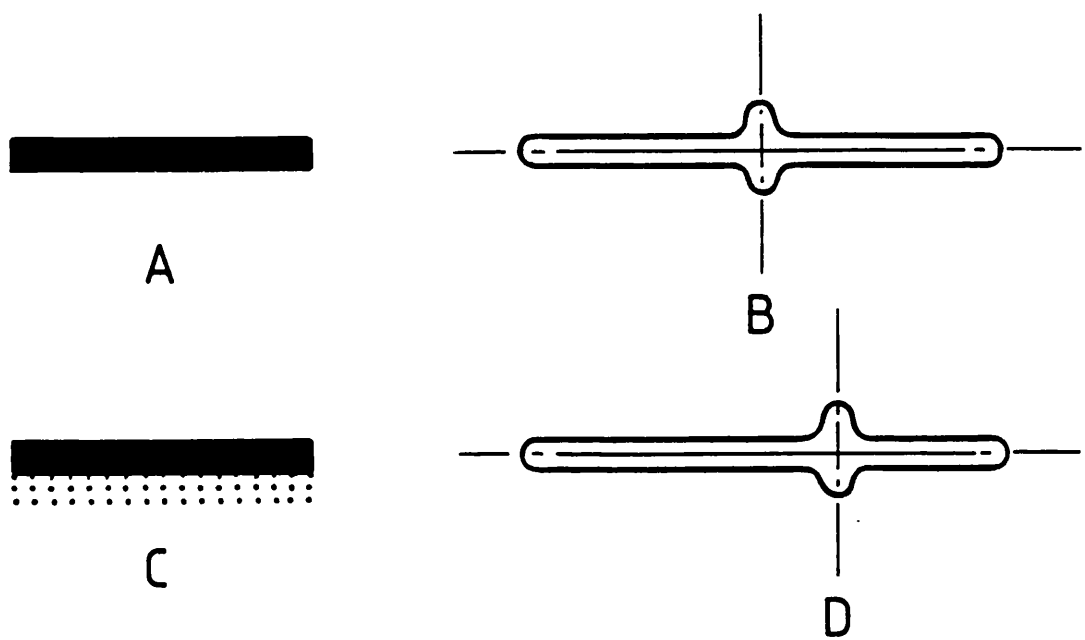


Fig. 4.4 a) Particle with sharp edges,  
b) histogram of angles for (a),  
c) particle with one diffuse edge,  
d) histogram of angles for (c).



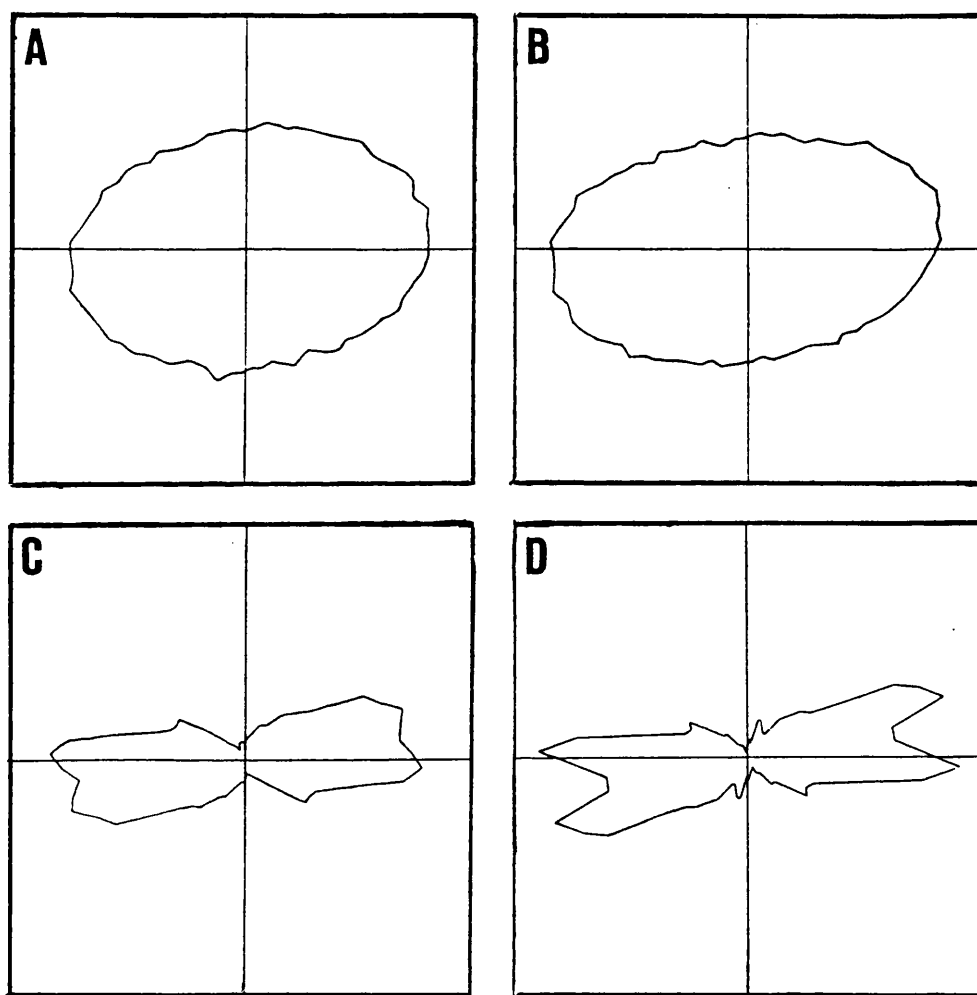


Fig. 4.5 Polar histograms calculated for Fig. 3.1a.  
a) unweighted histogram of angle of intensity gradient,  
b) weighted version of (a),  
c) unweighted histogram of mean direction,  
d) weighted version of (c),

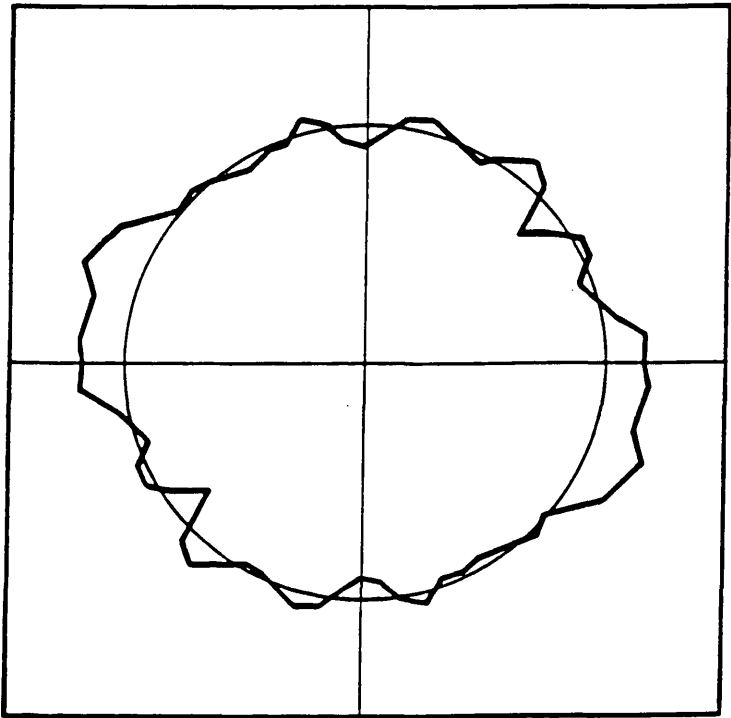


Fig. 4.6 Polar histogram of porosity of the image in Fig. 3.1a.

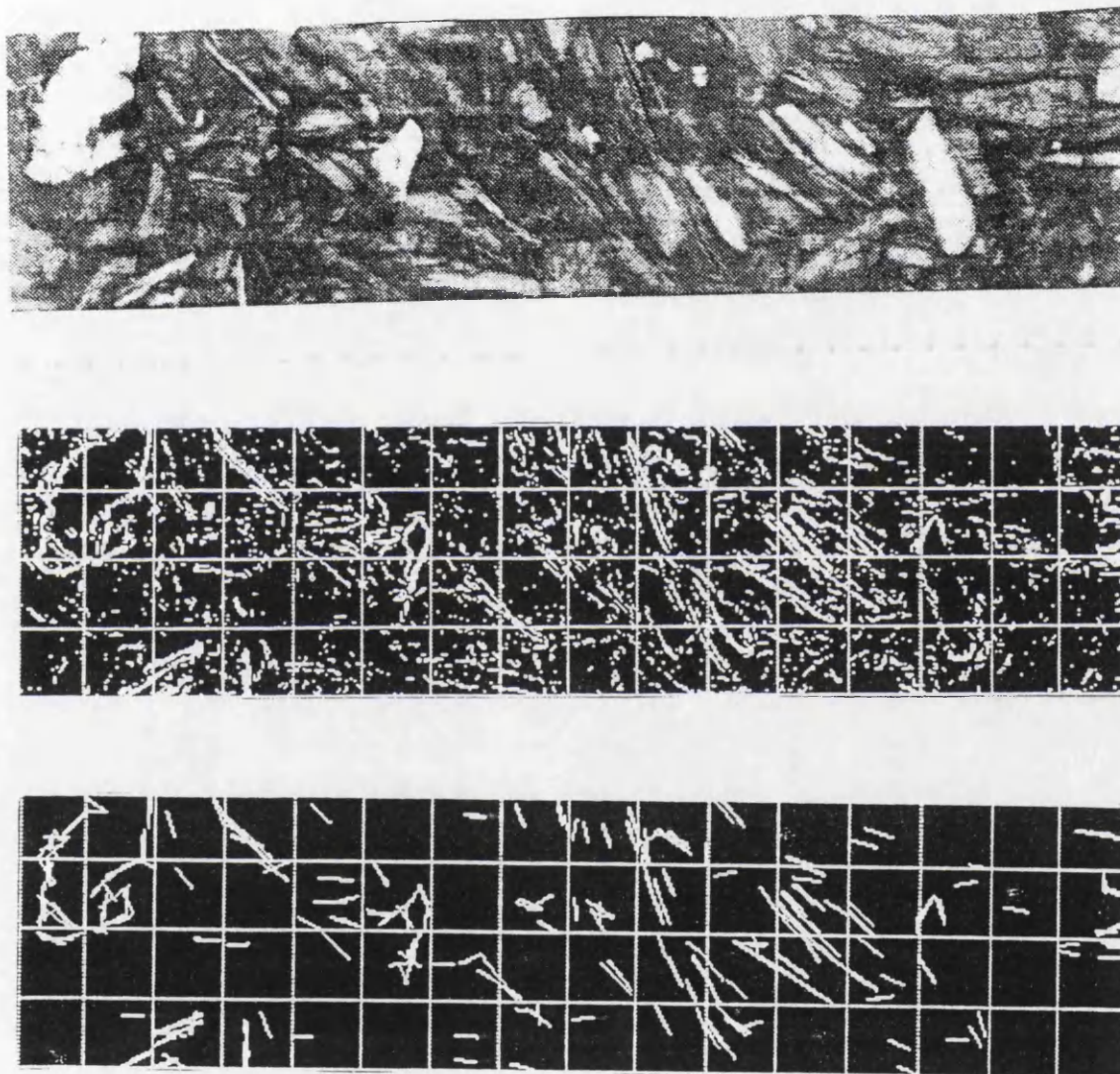


Fig. 4.7 a) Original image of a clay sample,  
b) detected edges of (a),  
c) detected straight line segments after Hough  
transform (Costa et al 1991a).

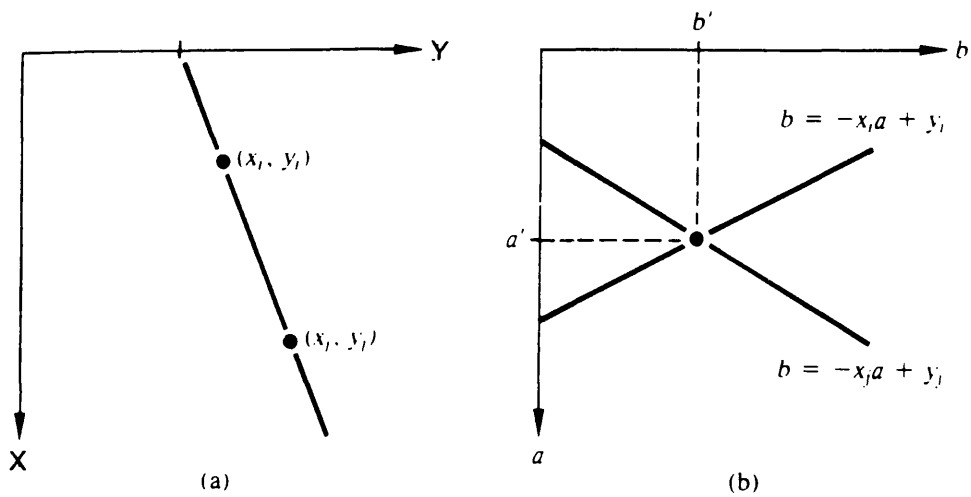


Fig. 4.8 a)  $x$ - $y$  plane, (b) Hough  $a$ - $b$  parameter space.

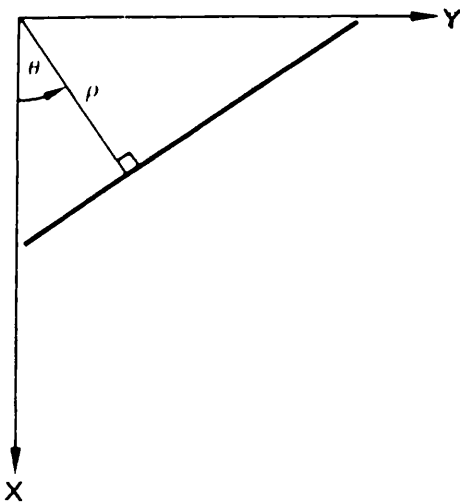
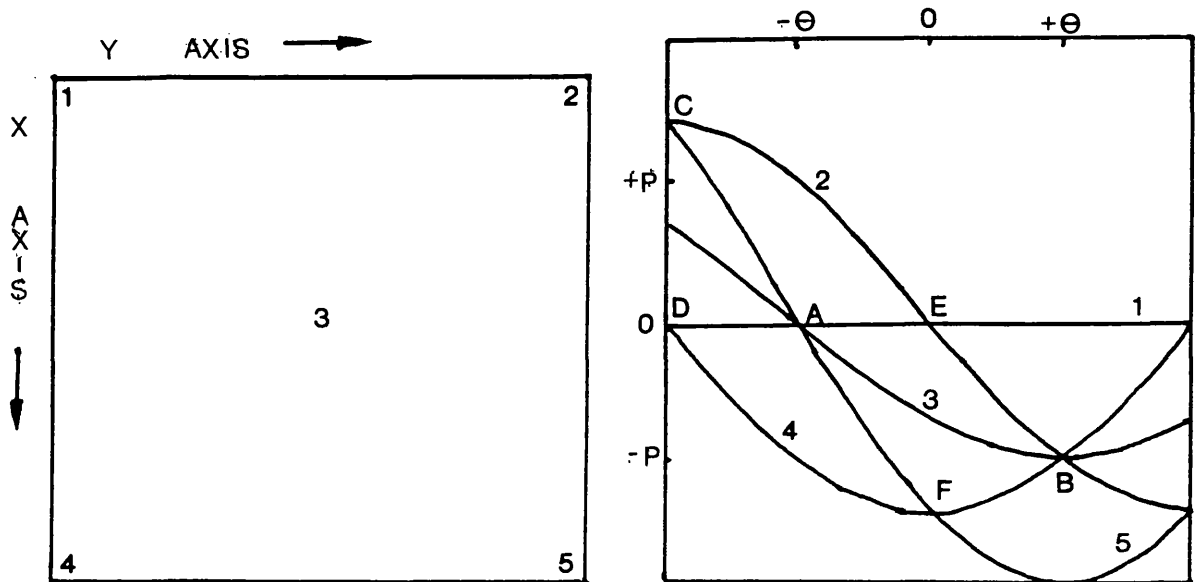


Fig. 4.9 a) Normal representation of a line,  
b) five point image,  
c) Hough  $\rho$ - $\theta$  parameter space.



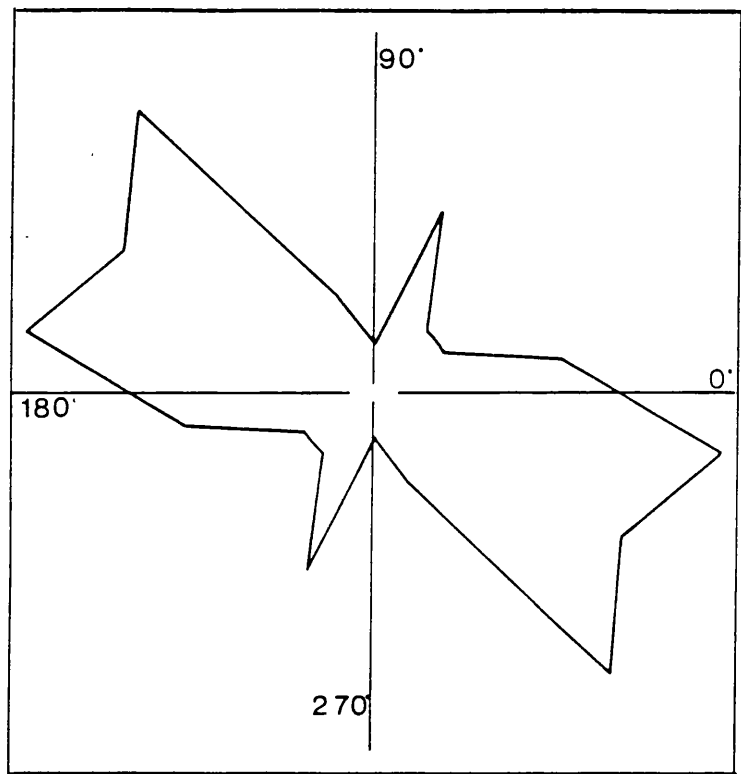


Fig. 4.10 The polar histogram for the orientation distribution of the straight line segments in the image of Fig. 4.5a (Costa et al 1991a).

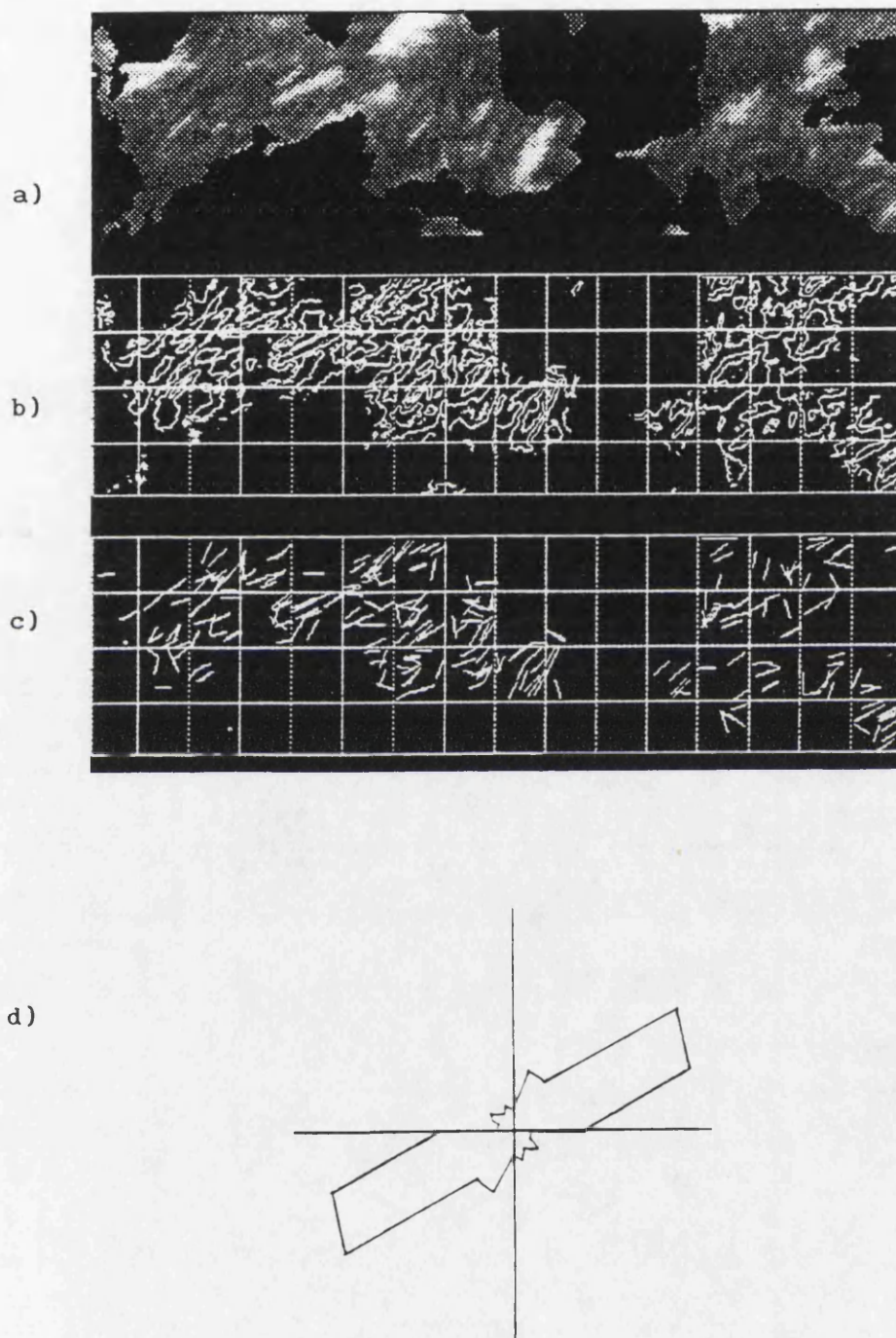


Fig. 4.11 Processing of the image corresponding to the domains with preferred orientation of  $45^\circ$  (Costa et al 1991b),  
 a) original picture,  
 b) detected edges,  
 c) reconstruction of (b) based on the detected straight line segments,  
 d) polar histogram of these straight lines of (c)

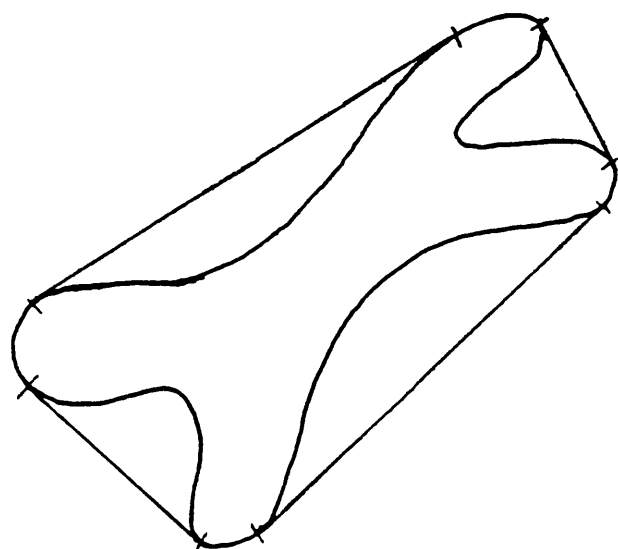


Fig. 4.12 Example of Convex Hull of a particle.



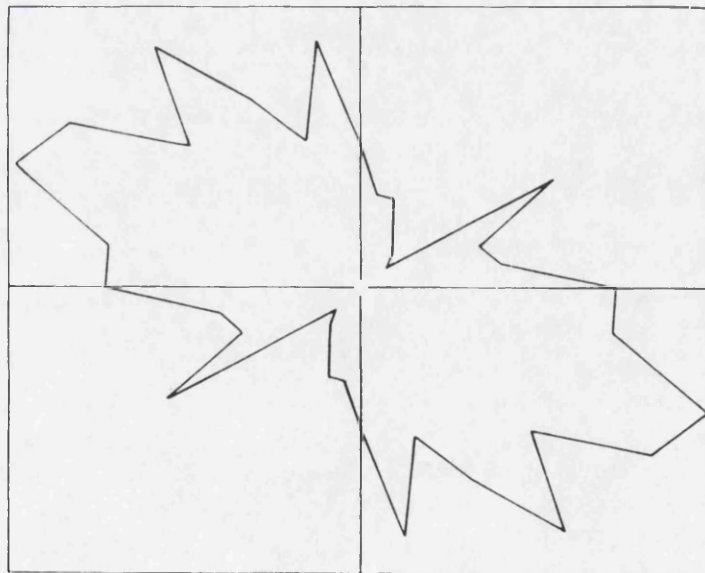
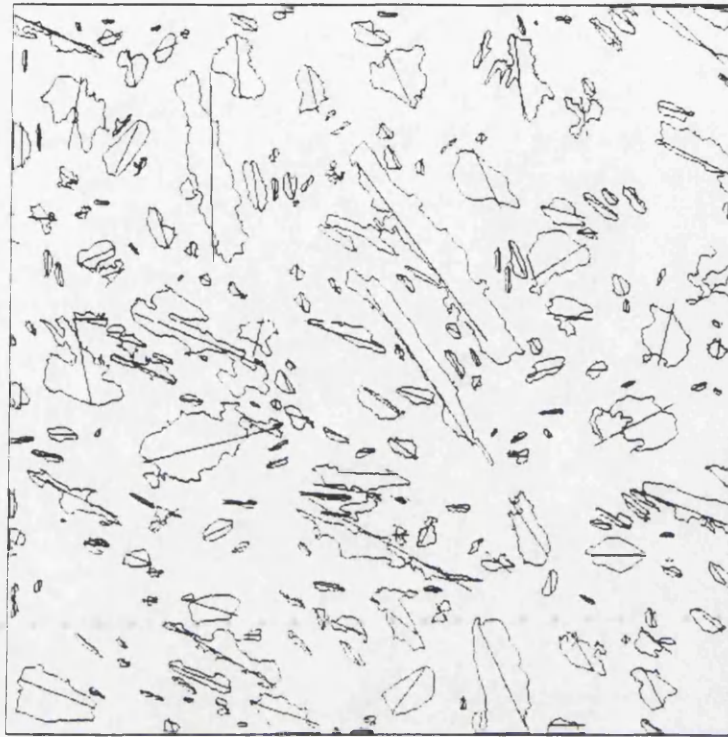


Fig. 4.13a) Experimental result by Directed Vein method  
(Lou et al 1991).

b) Polar histogram of the directed Vein in Fig. 4.12a  
(Lou et al 1991).



## CHAPTER 5 NON-LINEAR GREYSCALE MORPHOLOGY

### 5.1 INTRODUCTION

Mathematical morphology is a particular discipline in the field of image processing, which has been used to analyse the structures of materials in various fields such as mineralogy, petrography, angiography, cytology, etc. It was introduced in 1964, when Matheron started to investigate the relationships between the geometry of porous media and their permeabilities, and Serra was asked to quantify the petrography of iron ores in order to predict their milling properties. This initial research led to the formation of a team at the Paris School of Mines at Fontainebleau, the Centre de Morphologie Mathematique, which combined theoretical work with the design of practical applications, such as "texture analysis".

One of the basic intuitions of mathematical morphology is that the analysis of an image does not reduce to a simple measurement. Instead, it relies on a succession of operations which transform the image in order to make certain features apparent. These transformations mainly include erosion, dilation, opening, closing, skeletonization and so on.

Broadly speaking, mathematical morphology is an approach based on set-theoretic concepts. It has three aspects: 1) Algebraic, dealing with image transformation and derived from set-theoretical operations;

2) Probabilistic, concerning models of random sets applicable to the selection of small samples of materials; 3) Integral Geometrical, coping with image functions. (Heijmans and Sonse 1990)

A lot of research has been done to investigate the proper transforms on black and white (binary) images based on the boundaries of the objects. However, for some applications, grey-level transformations are needed (v.i. Section 5.2). Meyer (1978) and Sternberg (1982) were among the first to extend the mathematical morphology to grey-level images. A more thorough discussion, including some of the problems which arise if the set of grey-levels is finite, can be found in Ronse (1989). Heijmans and Sonse (1990) reviewed the previous work done in this field. In this Chapter, Section 5.2 gives the general theory and practice of binary morphology; Section 5.3 describes the development of grey-scale morphology; Sections 5.4, 5.5 and 5.6 explain the author's improvement on grey-scale morphology and its correction.

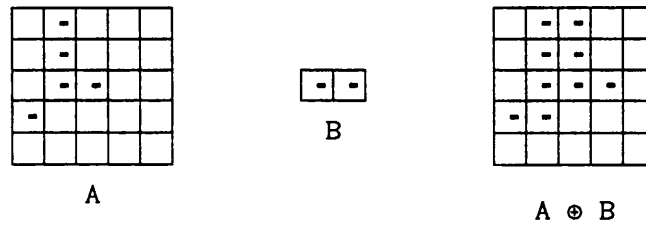
## 5.2 BINARY MORPHOLOGY

Mathematical morphology was first developed for the analysis of binary images, and its extension to grey-level images was a later development. Harlick et al. (1987) presented a tutorial, which may provide a quick understanding for beginning binary mathematical morphology. Normally, dilation is a morphological transformation which combines structure set  $B$  onto object set  $A$  by using vector addition of set elements. If  $A$  and  $B$  are sets in Euclidean  $N$ -space ( $E^N$ ) with elements  $a$  and  $b$  respectively,  $a=(a_1, \dots, a_n)$  and  $b=(b_1, \dots, b_n)$ , being  $N$ -tuples of element coordinates, then the dilation of  $A$  by  $B$  is the set of all possible vector sums of pairs of elements, one coming from

A and one coming from B (Harlick et al., 1987).

$$A \oplus B = \left\{ c \in E^N \mid c = a+b \text{ for some } a \in A \text{ and } b \in B \right\} \quad (5.1a)$$

An example of  $A \oplus B$  is shown in Fig. 5.1a.



$$A = \{(1,0), (1,1), (1,2), (2,2), (0,3)\}$$

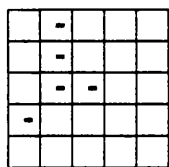
$$B = \{(0,0), (1,0)\}$$

$$A \oplus B = \{(1,0), (1,1), (1,2), (2,2), (0,3) \\ (2,0), (2,1), (2,2), (3,2), (1,3)\}$$

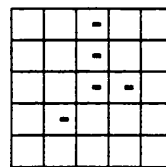
Fig. 5.1a The dilation of A by B

The interpretation can be done in a more convenient way of translation. If we define a translation of A by  $\mathbf{x}$ , which is denoted by  $(A)_{\mathbf{x}}$  here  $\mathbf{x}=(1,0)$  then the translated set is:

$$(A)_{(1,0)} = \{(2,0), (2,1), (2,2), (3,2), (1,3)\}$$



A

 $(A)_{(1,0)}$ 

$$A = \{(1,0), (1,1), (1,2), (2,2), (0,3)\}$$

$$x = (1,0)$$

$$(A)_{(1,0)} = \{(2,0), (2,1), (2,2), (3,2), (1,3)\}$$

Fig. 5.1b The translation of A by x

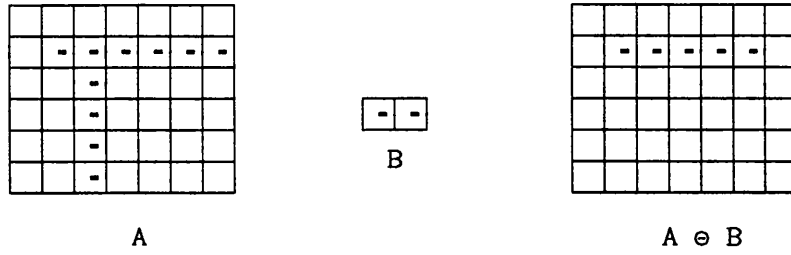
so the dilation of A by x is simply the union of the translations of A by all elements of  $A_x$  to form  $A \oplus A_x$ , so another way to present Equation 5.1a as

$$A \oplus B = \bigcup_b (A)_b \quad (5.1b)$$

Erosion is a morphological transformation dual to dilation, which combines two sets using the vector subtraction of set elements. If A and B are sets in Euclidean N-space, then the erosion of A by B is the set of all elements x for which  $x+b \in A$  for every  $b \in B$ . (Haralick et al 1987 p536)

$$A \ominus B = \left\{ x \in E^N \mid x+b \in A \text{ for every } b \in B \right\} \quad (5.2a)$$

An example of  $A \ominus B$  is shown in Fig. 5.2a.



$$A = \{(1,1), (2,1), (3,1), (4,1), (5,1), (6,1), (2,2), (2,3), (2,4), (2,5)\}$$

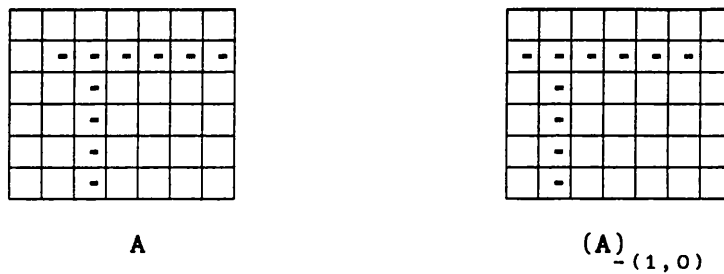
$$B = \{(0,0), (1,0)\}$$

$$A \ominus B = \{(0,1), (1,1), (2,1), (3,1), (4,1)\}$$

Fig. 5.2a The erosion of A by B

This can be interpreted in a more convenient way of translation. If we define a translation of A by  $-x$ , which is denoted as  $(A)_{-x}$  here  $x = (1,0)$  then

$$(A)_{-(1,0)} = \{(0,1), (1,1), (2,1), (3,1), (4,1), (5,1), (1,2), (1,3), (1,4), (1,5)\}$$



$$A = \{(1,1), (2,1), (3,1), (4,1), (5,1), (6,1), (2,2), (2,3), (2,4), (2,5)\}$$

$$x = (1,0)$$

$$(A)_{-(1,0)} = \{(0,1), (1,1), (2,1), (3,1), (4,1), (5,1), (1,2), (1,3), (1,4), (1,5)\}$$

Fig. 5.2b The translation of A by  $-x$

so the erosion of  $A$  by  $x$  can be expressed simply as the translations of  $A$  by all elements of  $A_{-x}$  to form  $A \ominus A_{-x}$  so another way to present Equation 5.2a could be:

$$A \ominus B = \bigcap_b (A)_{-b} \quad (5.2b)$$

Further,  $A$  dilated by  $B$  following erosion by  $B$  gives a closing by  $B$ ; and  $A$  eroded by  $B$  following dilation by  $B$  gives an opening by  $B$ . And several erosions on  $A$  by  $B$  (unless it would destroy connectivity of  $A$ ) may yield the medial axis of  $A$ , i.e. the skeleton, which is given as an example in this section.

In the author's practice, the structure elements,  $B$ , are normally circles with different radii, so the transformations of dilation and erosion are equivalent, respectively, to adding or removing a layer all around an object as shown in Fig. 5.3.

However, none of these transformations discussed above can be performed on the real images, unless the grey-level images have first been converted into binary images. Normally, the conversion could be done by choosing a threshold; the simplest threshold is a certain grey value. In an image, if the grey values of the pixels are larger than the threshold, then they will be turned to white (value of 255); and if the grey values of the pixels are smaller than the threshold, then they will be turned to black (value of 0). This conversion may either lose much important information of the original image or alter the structural features.

Fig. 5.4 gives an example of binary conversion based on different thresholds and then obtaining the skeleton of the object, which has two unsymmetrical sides. The third dimension expresses the grey level

of the object. The threshold is like a section cutting through the object in the horizontal; different thresholds result in different sizes of the binary area Fig. 5.4b and 5.4c. At the end of the process, the skeleton of the object will move towards the steep side of the object parallelly when the threshold is higher, Fig. 5.4d and 5.4e.

Fig. 5.5 is another example of doing the same operations on an object which has unsymmetrical ends Fig. 5.5a. The different thresholds give different shapes of binary areas Fig. 5.5b and 5.5c, which then cause the skeleton to turn in direction Fig. 5.5d and 5.5e.

From the above analysis, the simple grey level threshold not only moves the binary area of the object and its skeleton but also changes the direction of the skeleton. Adaptive thresholding can be used to solve these problems; however, in order to fully use the original information, greyscale morphology, which will not need the stage of converting to a binary image, has recently been developed. The morphological transformations of erosion and dilation can be performed on the original grey level image itself.

### 5.3 PREVIOUS GREYSCALE MORPHOLOGY

Greyscale morphological image processing originally treated the grey level function as sets or piles of binary cross sections (thresholds) and processing was based on a stereological approach in which cross sections were transformed individually or in pairs. This approach was later called "decomposition" of greyscale morphological structure elements. Different morphological structure elements were used separately, then the maximum value of the whole was chosen for

dilation, and the minimum value of the whole was chosen for erosion (Shih & Mitchell 1991).

Dyer and Rosenfeld (1979) developed an algorithm for grey level images that is a generalization of the standard thinning algorithms for binary images. In their algorithm, two points are connected if and only if the grey value of any element along the path that connects them is not less than both of them. This method defines the boundary point in a grey level image according to the grey level values of its neighbourhood, and then thins the image in a parallel mode, by substituting each boundary element with its minimum neighbour. Salari and Siy (1984) improved this approach and developed Ridge-Seeking of the object to obtain the skeleton of grey level images. This method requires that the pixels with grey values below a certain threshold are set to zero. A 3x3 pixels window is used to process the central pixel, N0, Fig. 5.6.

N4	N3	N2
N5	N0	N1
N6	N7	N8

Fig. 5.6 Pixel N0 and its eight neighbours.

N0 will be removed (changed to zero) if the following conditions are satisfied:

- 1) N0 is a boundary point (boundary obtained from the threshold);
- 2) N0 is not a local maximum point (not strictly greater than its neighbours);
- 3) N0 is not an end point (has more than one 4-neighbour nonzero);



- 4) NO's removal does not change the connectedness of its neighbours;
- 5) NO's removal does not weaken the strength of the 4-connectedness.

Other earlier methods of skeletonization of digital grey level images had been proposed. Levi and Montanari (1970) obtained the grey weighted distance transform (GWDT) of the image by associating to each point the length of the minimal path from that point to the background. The skeleton points are those that do not belong to the minimal path of any other points to the background. This method requires the digital picture to be segmented into black (background = zero) and white (object = non-zero).

Another approach called the min-max medial axis transformation (MMMAT) is based on the generalization of shrinking and expanding operations for grey level images (Peleg and Rosenfeld 1981). Local minimum and local maximum operations in a grey level image correspond to erosion and dilation operations of the binary picture, respectively (Nakagawa and Rosenfeld 1978).

#### 5.4 NON-LINEAR STRUCTURE FUNCTION

Heijmans and Ronse (1990) gave the underlying theory of grey-scale morphology as: if  $A$  and  $B$  are sets in Euclidean space ( $E^N$ ), then dilation and erosion are:

$$(A \oplus B)(x) = \sup_{h \in E} (A(x-h) + B(h)) \quad (5.3)$$

$$(A \ominus B)(x) = \inf_{h \in E} (A(x+h) - B(h)) \quad (5.4)$$

Here,  $A(x)$  is the function on which the dilation and erosion are operated and  $B(x)$  is the structure function. The operations are shown in Fig. 5.7.

However, all the previous greyscale morphology has been based on a linear structure element  $B$ . Even though the object set  $A$  has been extended from a binary image to a grey level image described by  $A(x)$ , and the structure element has been improved to the structure function  $B(x)$ , this structure function is always the same over the whole image, for example see Fig. 5.7.

For some real images, such as the scanning electron micrographs of soil on which the author operates, the edges of the particles are very fuzzy, or for some reasons quite a few of the particles have grey values much lower than that which had been expected. In this research, these fuzzy and quite dark small particles are sometimes really important and cannot be ignored. In this case, it will be difficult to get the correct result by using binary morphological transformations after grey value thresholding. In order to transform equally on all the edges of both brighter or darker particles, greyscale morphological transformations have been chosen in this research.

However, in the work done by Heijmans and Sonse (1990), Fig. 5.7, if  $A(x)$  is supposed to be the profile of a particle, and  $B(x)$  is supposed to be the structure function for the transformation, then the void could be filled up or become very unclear after dilation, and the particles could be taken off or become very unclear after erosion. This guides the author to give a further consideration to the structure function  $B(x)$ . The linear structure function should be

replaced by a non-linear structure function, so that the steep edges will be eroded and dilated more than the gentle edges, and for the top parts of the particles and the bottom parts of the voids the transforms will be affected only a little.

The idea of choosing the magnitude of the intensity gradient at each pixel of the image as the structure function of non-linear greyscale morphology arose from the above analysis. The author also considered using  $\text{lap } I$  i.e.  $I_{\text{new}} = f(I_{\text{old}}, \text{mod grad } I, \text{lap } I)$ , but the use of  $\text{mod grad } I$  alone was sufficient for the work described below. Because the gradient of the grey level is just in accord with the demands of the structure function which the author wants, the steep edges have bigger gradients and gentle edges have smaller gradients. Therefore, when eroding the steeper edges, the grey value will be decreased more and the edge will be cut back more; on the gentle edges, the grey value will be decreased less; and the on flat top areas of particles or the flat bottom areas of voids, having gradient values of zero, the grey value will be unaltered and the erosion will not operate. When doing dilation on the steep edges, the grey value will be increased more, and the edge will be expanded more; on the gentle edges, the grey value will be increased less and the edge will be expanded less. Fig. 5.8 illustrates erosion and dilation operating on a profile of the feature in Fig. 5.4. The structure function, gradient is calculated as:

$$U(i) = I(i) - I(i-1), \quad (5.5)$$

where,  $U(i)$  is the value of structure function, i.e. the gradient,  
at point  $i$ ,

$I(i)$  is the grey value at point  $i$ .

So the transformed grey values of  $I(i)$  are:

$$\text{erosion:} \quad I_e(i) = I(i) - kU(i); \quad (5.6)$$

$$\text{dilation:} \quad I_d(i) = I(i) + kU(i). \quad (5.7)$$

Where  $I_e(i)$ : the eroded result,

$I_d(i)$ : the dilated result,

$k$ : the fraction of transformation.

Note, only absolute value of  $U(i)$  had been used here.

Fig. 5.8a is a profile of the feature in Fig. 5.4a; Fig. 5.8b shows the gradient values on the profile; Figs. 5.8c and 5.8d are the eroded results and dilated results compared with the broken line of the original profile respectively, with fraction value of 1 in eqn. 5.7.

For a two-dimensional image,  $I$ , the grey level of each pixel is the third dimension  $I(i,j)$ . The 20,14-formula (Smart and Tovey 1988) for calculating the intensity gradient at each pixel has been employed here (see Section 3.2). The coefficients for calculating  $dx$  and  $dy$  are shown in Equations (3.5) and (3.6) respectively: and the two-dimensional gradient  $U(i,j)$  can be calculated as:

$$U(i,j) = \sqrt{dx^2 + dy^2} \quad (5.8)$$

So, the transformed grey values of  $I(i,j)$  equal:

$$\text{erosion:} \quad I_e(i,j) = I(i,j) - kU(i,j) \quad (5.9)$$

$$\text{dilation: } I_d(i,j) = I(i,j) + kU(i,j) \quad (5.10)$$

A series of operations of two-dimensional grey-level erosions and dilations have been carried out on the same feature as Fig. 5.4. and an illustrated in Fig. 5.9. This figure shows three photographs of the screen at successive stages of the process. In each photograph, the original image is shown at top left and the original 'grey scale cross-section' at top right. The partly transformed result after a few cycles of erosion is shown at centre left; and the successive cross-sections are shown at centre right. The partly transformed result and the corresponding cross-sections after several more cycles are show at the bottom.

### 5.5 THE CUBIC CURVE CORRECTION

Sampling and digitizing real images often reduce the quality of the images, since the particles are not bright enough (it was hoped that features have grey value of 255 at the top), and the voids are not dark enough (it was hoped that features have grey value of 0 at the bottom). After normal greyscale transformation, the contrast will stay the same or even smaller than before. There are many existing methods to increase the contrast of images. In the author's experiment, it was desired to increase the contrast whilst keeping the basic structure of the image, so the cubic curve correction has therefore been employed in this work.

In Fig. 5.10, the X axis shows the grey value,  $x$ , before correction, and the Y axis shows the grey value,  $y$ , after correction. By using the cubic curve correction,  $y=ax^3+bx^2+cx$ , the grey value will

be increased when it is bigger than a certain value,  $t$ ; and the grey value will be decreased when it is smaller than  $t$ ; at the point of  $(t, t)$  the value will be unaltered by the correction. The maximum value,  $m$ , which equals 255, is not changed after correction; and the minimum value, which equals 0, is not changed either. These conditions may be expressed explicitly as:

$$\begin{array}{lll} \text{thus} & x = 0 & y = 0; \\ & x < t & 0 < y < x; \\ & x = t & y = t; \\ & x > t & x < y < m; \\ & x = m & y = m. \end{array}$$

$$\text{So for the curve} \quad y = ax^3 + bx^2 + cx \quad (5.11)$$

$$\text{has:} \quad \begin{cases} m = am^3 + bm^2 + cm \\ t = at^3 + bt^2 + ct \end{cases}$$

$$\text{or} \quad \begin{cases} am^3 + bm^2 + (c-1)m = 0 \\ at^3 + bt^2 + (c-1)t = 0 \end{cases}$$

$$\text{therefore:} \quad \begin{cases} am^2t + bmt + (c-1)t = 0 \\ at^2m + btm + (c-1)m = 0 \end{cases}$$

$$\text{so:} \quad amt(m-t) + (c-1)(t-m) = 0$$

$$a = \frac{c-1}{mt} \quad (5.12)$$

$$\begin{aligned}\frac{c-1}{mt} m^2 + bm + (c-1) &= 0 \\ bm &= (1-c) \frac{m+t}{t} \\ b &= \frac{(1-c)(m+t)}{mt}\end{aligned}\quad (5.13)$$

Now the curve become:

$$y = \frac{c-1}{mt} x^3 - \frac{(m+t)(c-1)}{mt} x^2 + cx \quad (5.14)$$

with  $t$  and  $c$  to be determined.

Combining the cubic curve correction. Equation. (5.14), with the grey level transformation of Equations (5.11 and 5.12):

$$\begin{aligned}I_e(i, j) &= I(i, j) - kU(i, j) \\ I_d(i, j) &= I(i, j) + kU(i, j)\end{aligned}$$

the correction of the transformed image is:

$$\begin{aligned}I_e'(i, j) &= \frac{c-1}{mt} (I(i, j) - kU(i, j))^3 - \frac{(m+t)(c-1)}{mt} (I(i, j) - kU(i, j))^2 + \\ &+ c(I(i, j) - kU(i, j))\end{aligned}\quad (5.17)$$

$$\begin{aligned}I_d'(i, j) &= \frac{c-1}{mt} (I(i, j) + kU(i, j))^3 - \frac{(m+t)(c-1)}{mt} (I(i, j) + kU(i, j))^2 + \\ &+ c(I(i, j) + kU(i, j))\end{aligned}\quad (5.18)$$

where  $I_e(i,j)$ ,  $I_d(i,j)$  are the grey values after erosion and dilation,  
 $I'_e(i,j)$ ,  $I'_d(i,j)$  are the grey values after cubic curve  
 correction.

In the cubic curve correction, there are still two independent parameters,  $t$  and  $c$ ; and it will be necessary to adjust these to suit the images which are to be processed. Here follows a simple example to choose one normal set of parameters. First the author set

$$t = m/2$$

then (5.14) becomes:

$$y = \frac{2(c-1)}{m^2} x^3 - \frac{3}{m} (c-1)x^2 + cx .$$

Now consider:  $x = m/4$

$$y = \frac{3c+5}{32} m$$

Now for  $x = m/4 < t$ , we require  $y < x = m/4$

so: 
$$y = \frac{3c+5}{32} m < \frac{m}{4} = \frac{8}{32} m$$

therefore:

$$c < 1$$

For this example considered above the parameters have been set as:

$$c = 0.5$$

$$t = m/2$$

$$m = 255$$



finally, the cubic curve correction with this set of parameter become:

$$y = -\frac{x^3}{m^2} + \frac{3x^2}{2m} + \frac{x}{2} \quad (5.19)$$

This curve is plotted in Fig. 5.10a. Finally, correcting eroded and dilated images respectively:

$$I_e'(i,j) = -\frac{I_e^3(i,j)}{m^2} + \frac{3I_e^2(i,j)}{2m} + \frac{I_e(i,j)}{2} \quad (5.20)$$

$$I_d'(i,j) = -\frac{I_d^3(i,j)}{m^2} + \frac{3I_d^2(i,j)}{2m} + \frac{I_d(i,j)}{2} \quad (5.21)$$

## 5.6 LOOK-UP-TABLE AND PRACTICE

In practice, grey scale morphological transformation of images means evaluating formulae 5.17 and 5.18 for each pixel. Direct calculation would take a very long time to do only one transform. In the author's code, a Look Up Table of grey value against intensity gradient was used. The third dimension, Z, is the grey value of the pixel after transformation, X is the original grey value, and Y is the intensity gradient. Fig. 5.11 shows the Erosion Look Up Table, which was used for Fig. 5.12. For the scanning electron micrograph of clay in Fig. 5.12a, the parameters have been set as:

$$c = 0.5,$$

$$t = m/4,$$

$$m = 255.$$

$$k = \frac{1}{2}$$

So, the formula used for this process is:

$$I'_e(i,j) = \frac{2}{m^2} \left( I(i,j) - \frac{1}{2} U(i,j) \right)^3 - \frac{5}{2m} \left( I(i,j) - \frac{1}{2} U(i,j) \right)^2 + \frac{1}{2} \left( I(i,j) - \frac{1}{2} U(i,j) \right) \quad (5.22)$$

In the author's code when  $I'_e$  is larger than 255, it is put equal to 255; and when  $I'_e$  is smaller than 0, it is put equal to 0. So when  $U(i,j)$  equals 0, the correction curve is as shown in Fig. 5.10b.

Fig. 5.12 shows an example of the process of skeletonization. Fig. 5.12a is a scanning electron micrograph of clay with 2,000x magnification. Fig. 5.12b is an early stage of skeletonization, at which the program has picked up the particles which need to be processed, and some very fine particles have been skeletonized. Fig. 5.12c shows a mild stage of skeletonization, at which the mid-size particles have been skeletonized as well. Fig. 5.12d shows a late stage of skeletonization, at which most of the particles have been skeletonized, and only a few very large particles still need to be skeletonized. Fig. 5.12e is the final stage of the process, which is the skeleton of the micrograph shown in Fig. 5.12a. From the whole process, we can see that, when the thin particles have been eroded into single lines, the erosion will not affect these skeletons any more; the erosion will act only on the fat particles, until they have been skeletonized too.

In Fig. 5.12e, the skeletons sometimes are bent, caused by joining of the ends of two particles lying in different directions; or

sometimes for one particle, the skeleton may become discontinuous because of variability of the grey level on the ridge of the particle. However, all these minor problems can be easily solved by other image analysis methods, such as the Hough transform which was discussed in Chapter 4.

## 5.7 CONCLUSION

This new algorithm of non-linear greyscale morphology has been proved both in theory and in practice; it evades one of the most difficult stages in image processing, *i.e.* converting to a binary image. It treats particles of different brightness and different size particles equally. This method not only extends the operation from binary image to greyscale image; but it also employs the non-linear structure function for transformation and combines this with the cubic curve correction. Although there are some parameters which need to be set according to the characteristics of different images, the calculations which has been made in this research have given a very successful result in Fig. 5.12.

There are many methods of improving the skeleton of the images. It could be possible, for example, Hough transform could be used both to join two lines in the same direction with one pixel gap in between, and to cut the joinpoint of lines in different directions. The simplest method of cutting the joinpoint is to measure whether there are more than two neighbour points in the 3x3 mask not lying on the same line, while the middle point is lying on a line.

There are some further applications which could be made. The immediate application of this algorithm is using the skeletonized

image for the Hough transform and the Method of Veins (Chapter 4) instead of using the edges of particles, and this will give a much more accurate result. Also the skeletonized image can be used to count isolated points, the ends of particles, and the joins of particles, in order to classify the images. And the skeleton of voids can be easily obtained to analyse the distribution of the voids. The main further application of the skeletons could be used to study the microstructure of individual particles such as length and size distribution, contact angle, and orientation.

In the course of these experiments, it was noted that, by taking a proper choice of parameters, either Equation 5.14, or Equation 5.20 could be arranged to produce a black-and -white image. This might be a new way of converting to a binary image. This binary image could then be used to do a lot of analysis, such as analyzing the void ratio of the image.

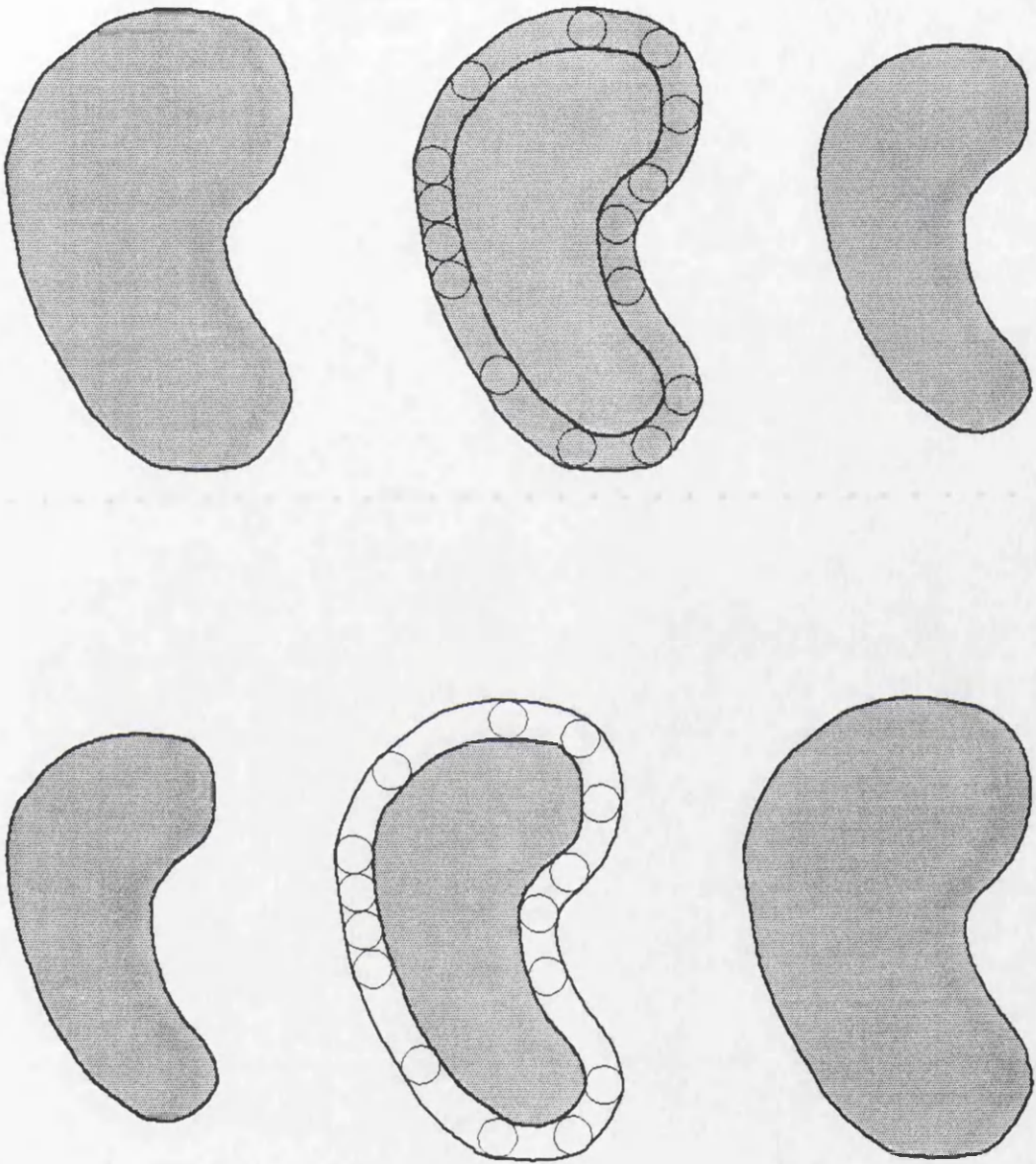


Fig. 5.3 a) Binary transform of erosion

b) Binary transform of dilation

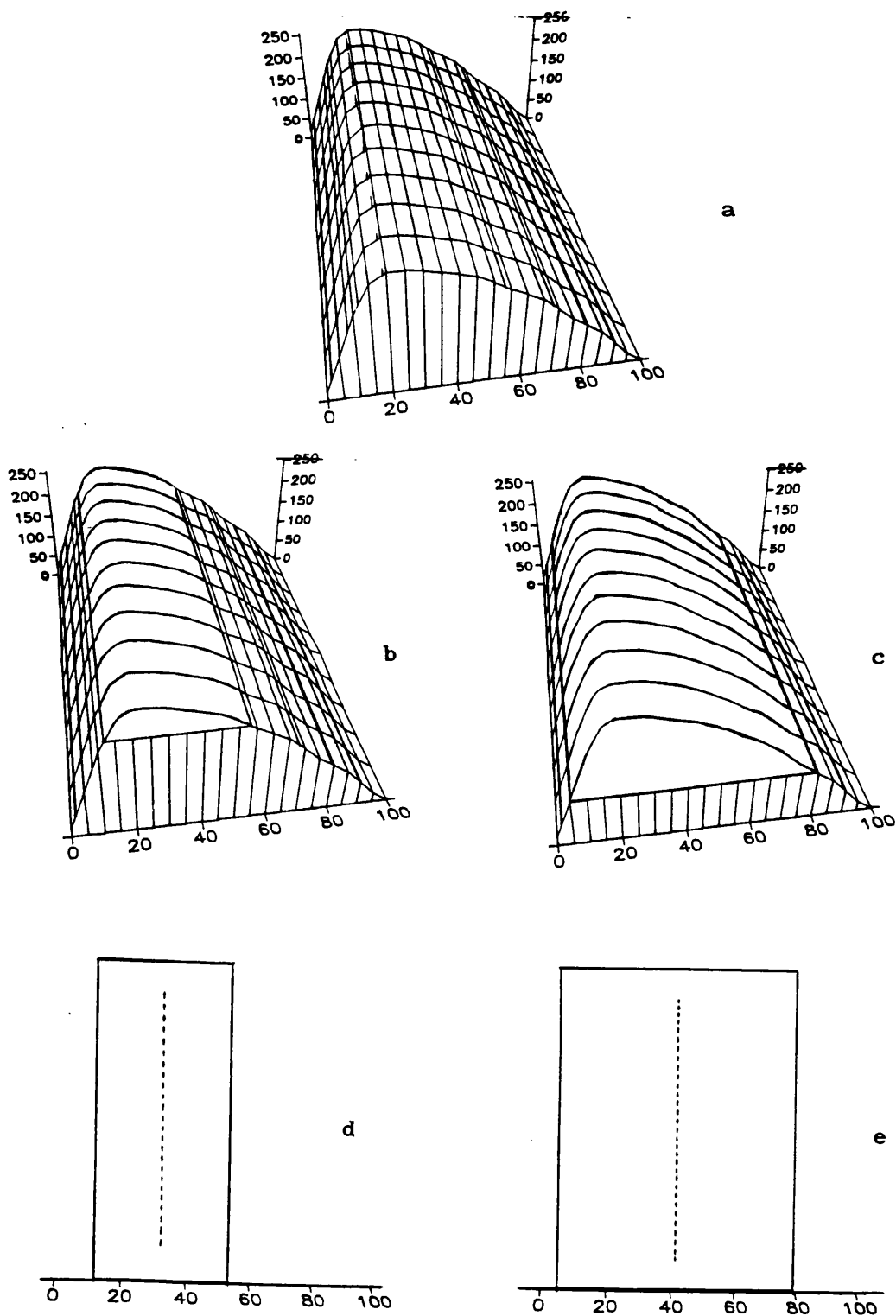


Fig. 5.4 a) Feature with unsymmetrical sides;  
b) feature has been cut by choosing grey level threshold of 200;  
c) feature has been cut by choosing grey level threshold of 100;  
d) transformed binary feature and its skeleton  
after process of (b);  
e) transformed binary feature and its skeleton  
after process of (c).

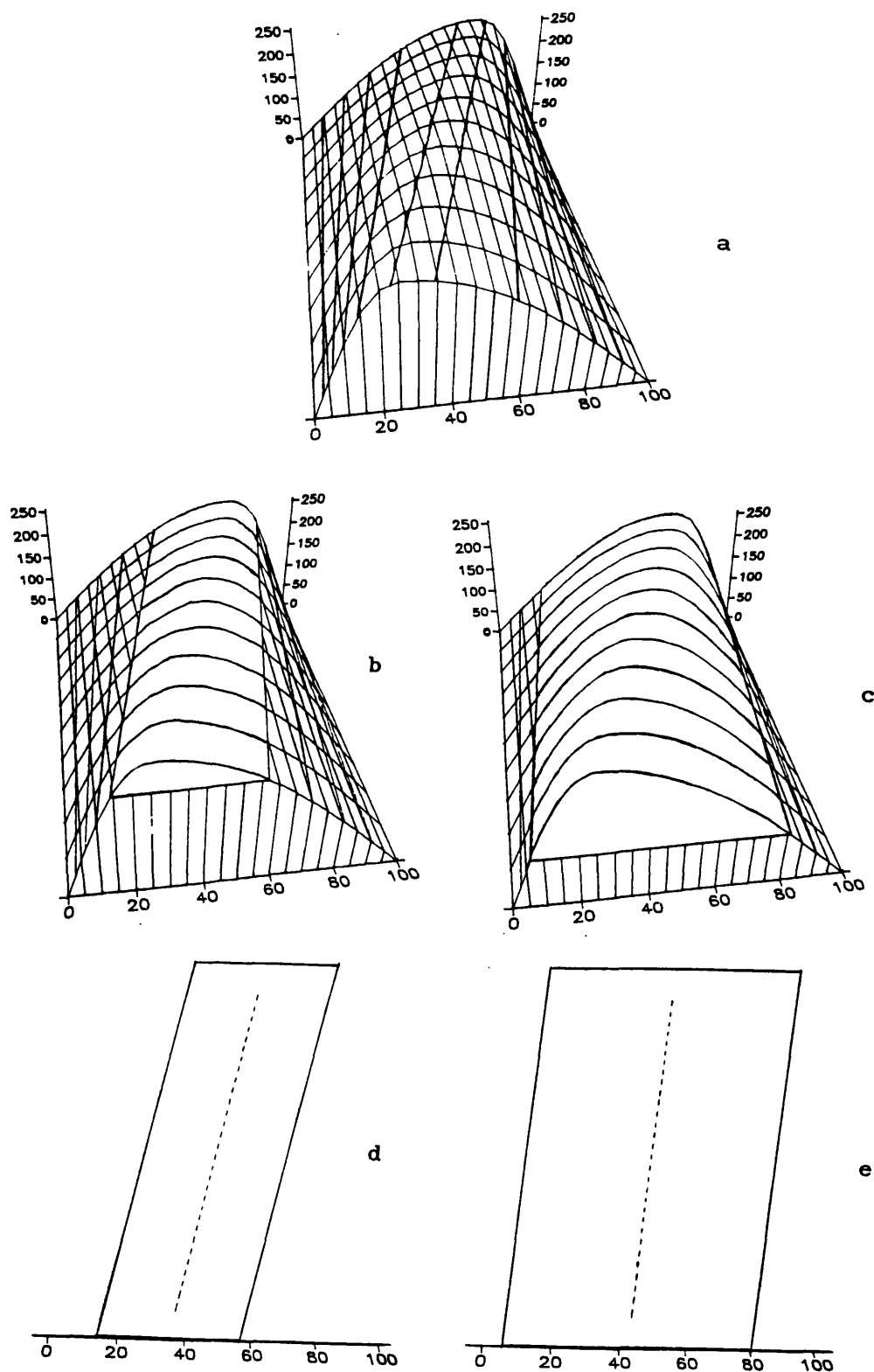


Fig. 5.5 a) Feature with unsymmetrical sides;  
 b) feature has been cut by choosing grey level threshold of 200;  
 c) feature has been cut by choosing grey level threshold of 100;  
 d) transformed binary feature and its skeleton  
 after process of (b);  
 e) transformed binary feature and its skeleton  
 after process of (c).

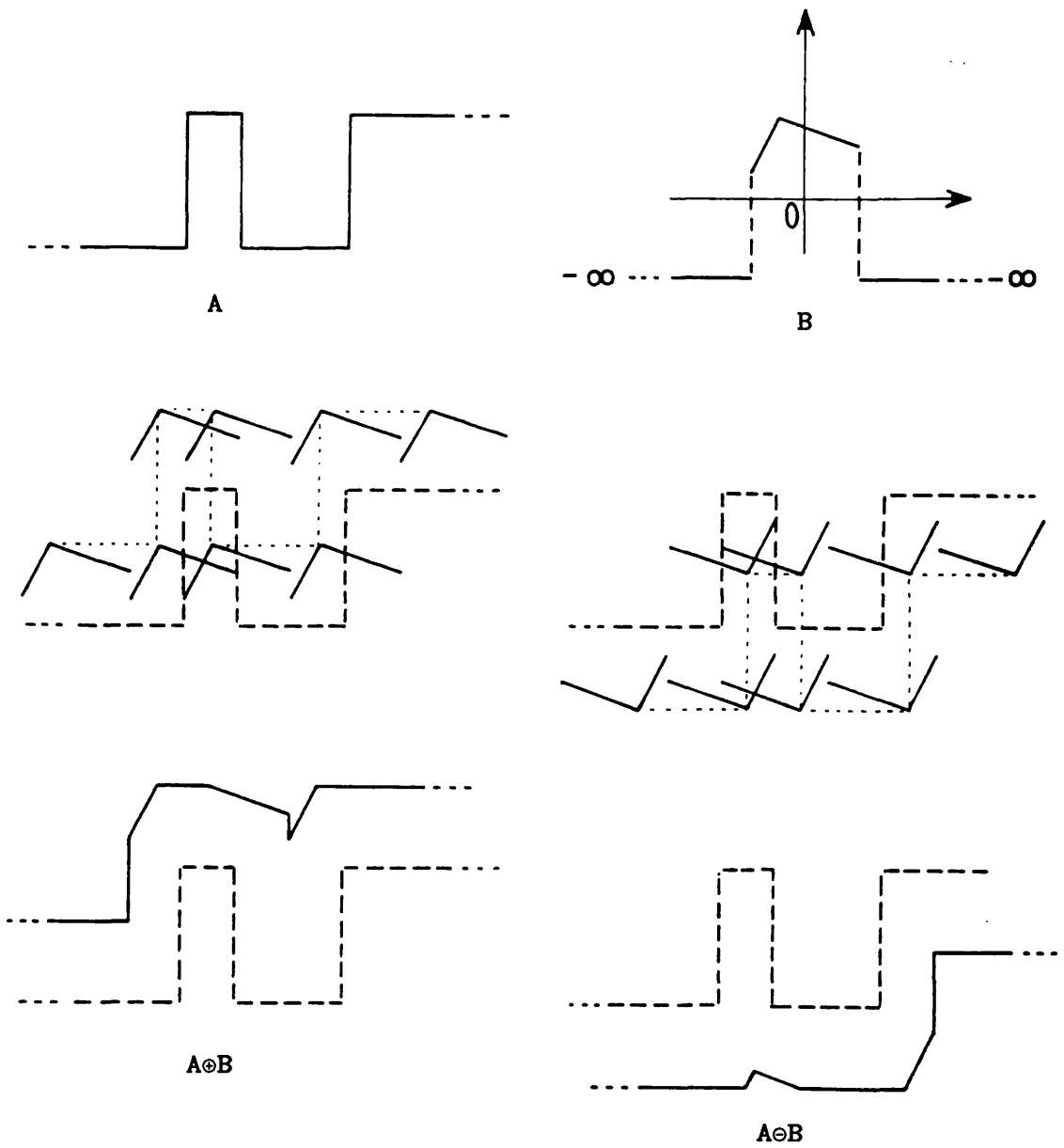


Fig. 5.7 Dilation and erosion of a function  $A(x)$  by a structure function  $B(x)$  (Heijmans & Ronse 1990).



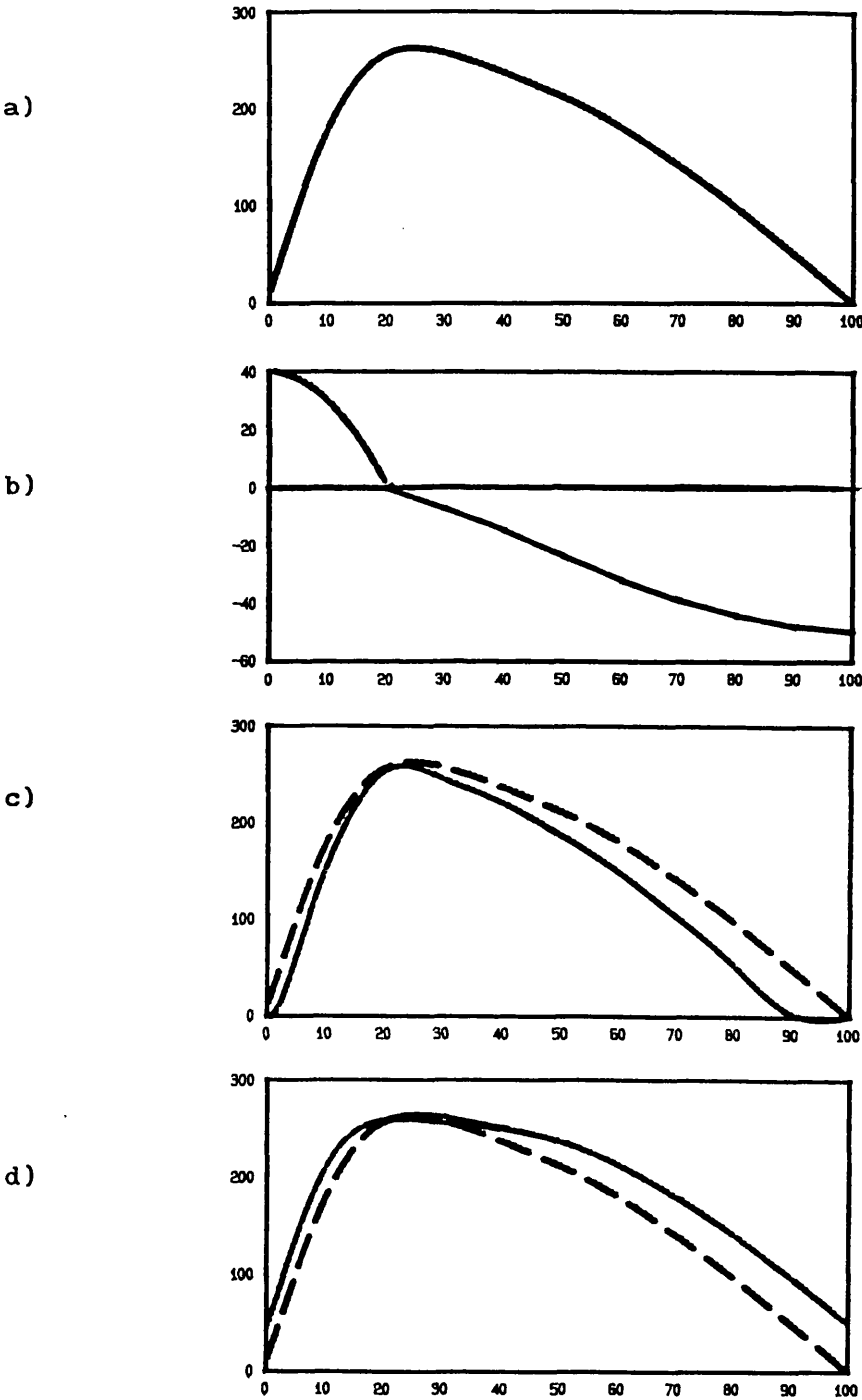


Fig. 5.8a) One cross section of the feature in Fig 5.4;  
b) gradient value at each point of (a);  
c) result of eroded transform;  
d) result of dilated transform.

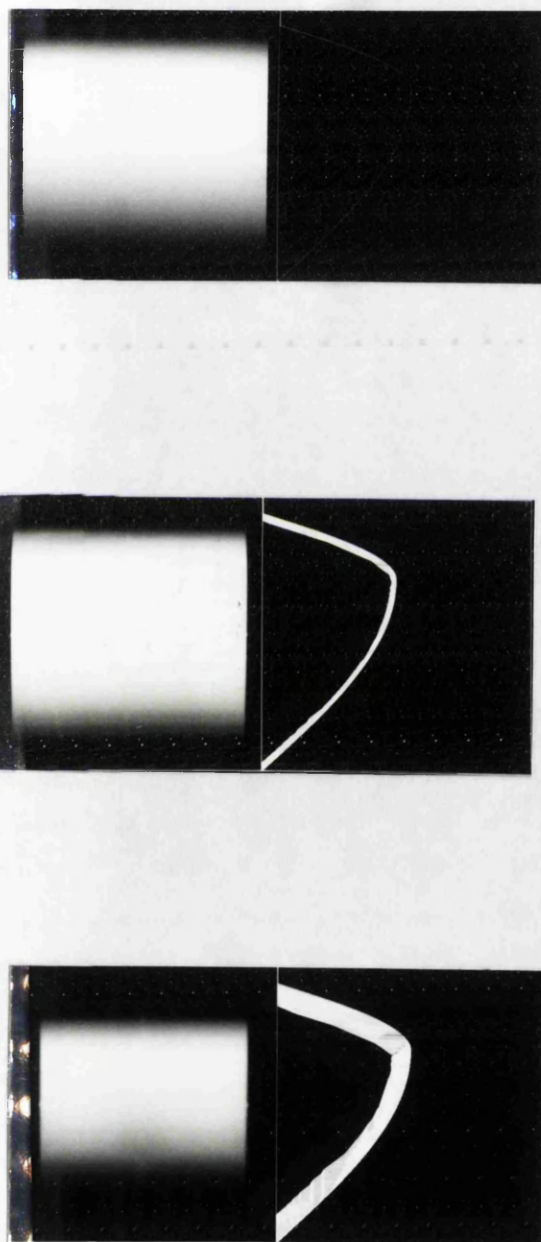


Fig. 5.9 Process of two-dimensional erosion on the feature in Fig. 5.4, left: the image; right: successive cross-section of the image; top: at start; below: after repeated erosion.

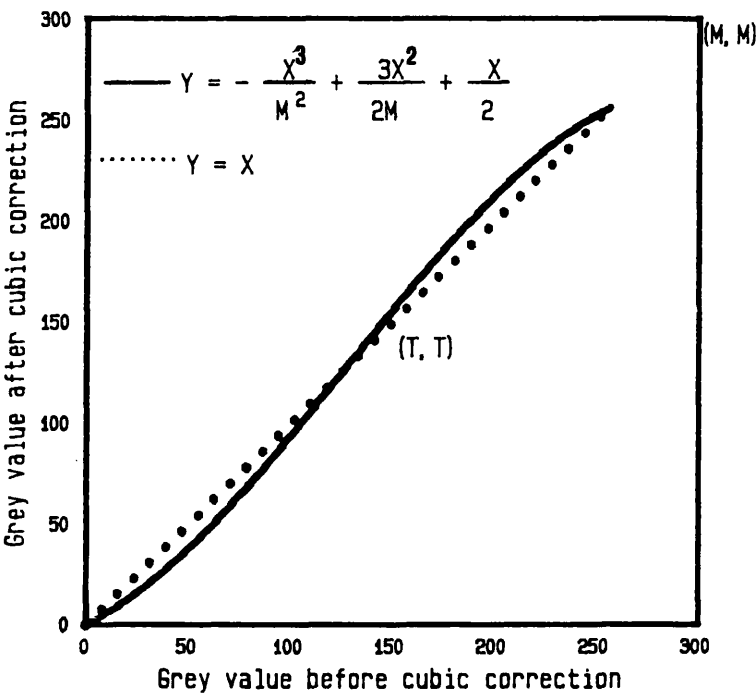


Fig. 5.10a Cubic correction curve with  $t=m/2$ ;  $m=255$ ; and  $c=0.5$ .

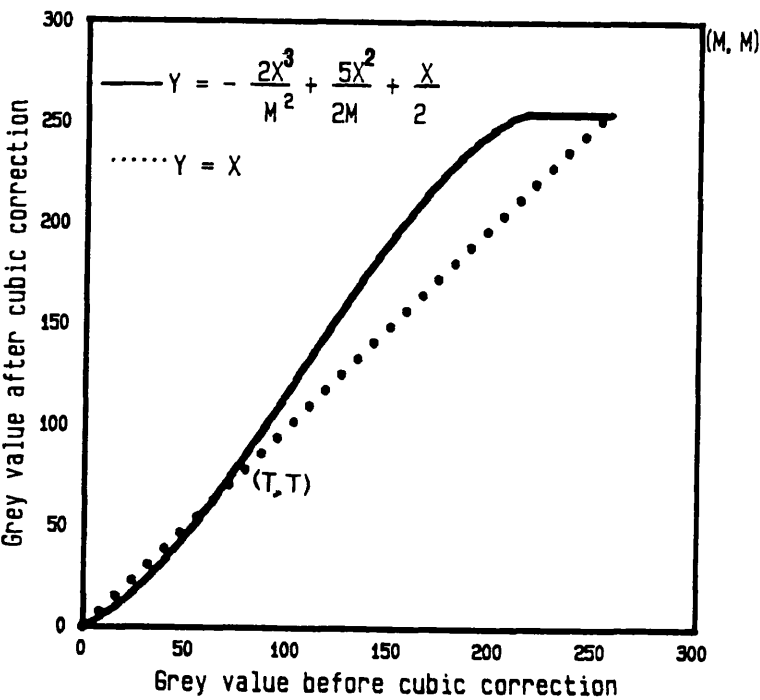


Fig. 5.10b Cubic correction curve with  $t=m/4$ ;  $m=255$ ; and  $c=0.5$ .

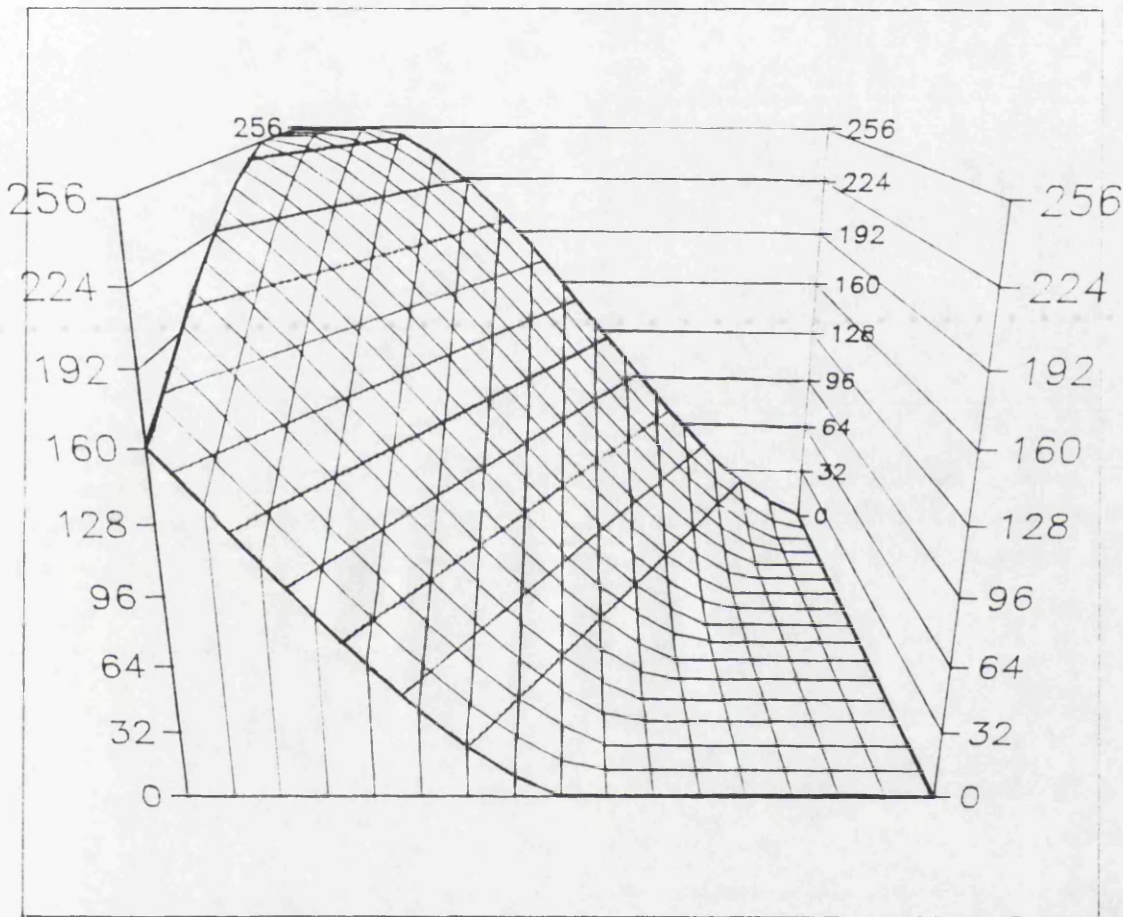


Fig. 5.11 Erosion Look Up Table.

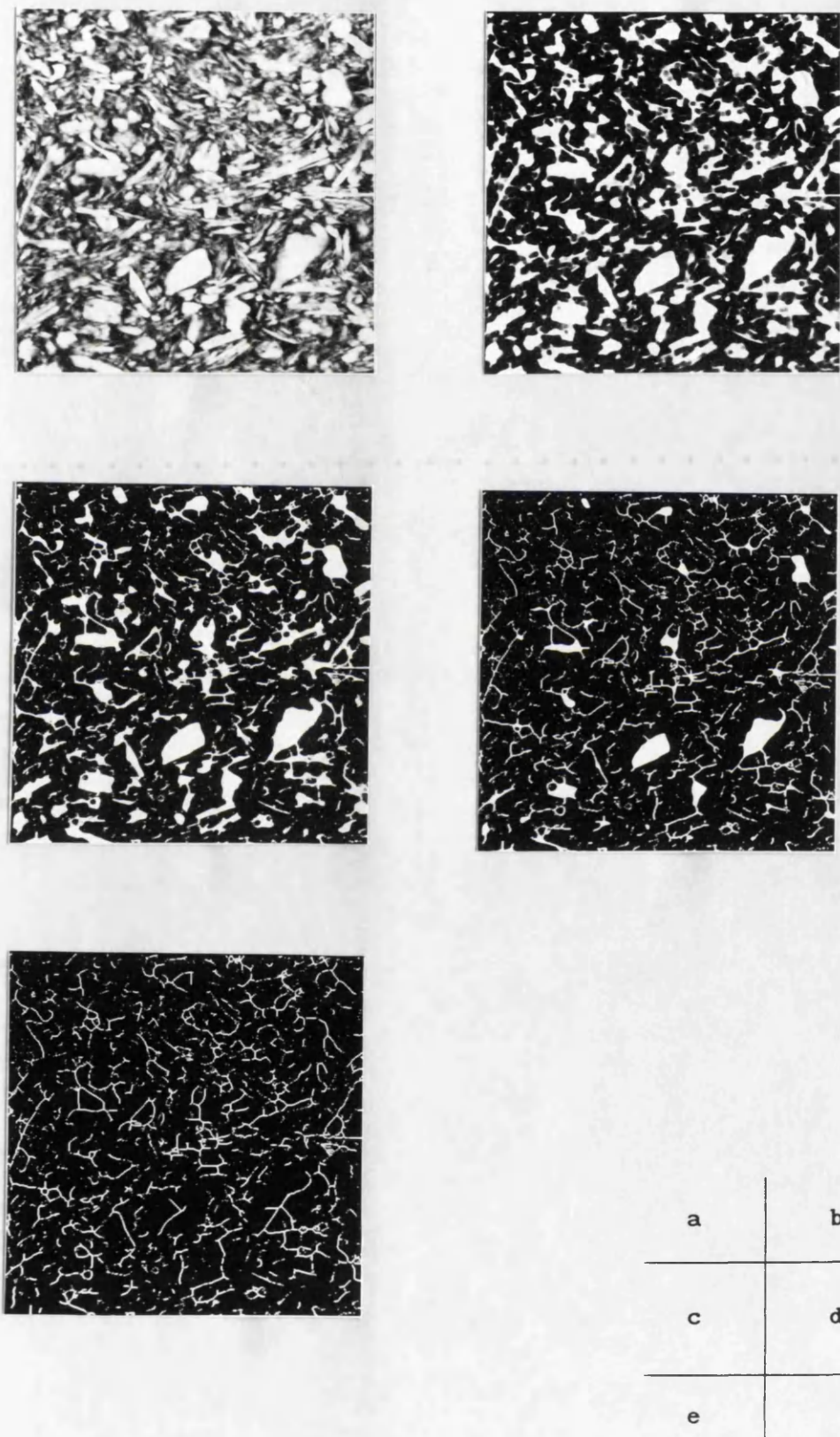


Fig. 5.12 The process of skeletonization from a through b, c, d, to e.

PART TWO PARALLEL PROCESSING

## CHAPTER 6 PARALLEL PROCESSING — PRINCIPLES

### 6.1 INTRODUCTION

*PARALLELISM -- many processors working together, all simultaneously executing some portions of a procedure for solving a problem.*

Instead of using a single processor which does one operation after another in sequence, the idea of dividing one computing task into subtasks, which need not be executed sequentially but could be executed simultaneously in parallel, has been around for at least 40 years. However, many reasons hindered the development of parallel computing.

Over the last few years, in particular by using transputers, the parallel approach has established itself as one of the best ways to make computers fast enough and powerful enough to satisfy the demands of scientists and engineers.

### 6.2. LEVEL OF PARALLELISM

To realize parallelism, first the task has to be divided into subtasks according to:

- 1) what kind of problem needs to be solved;

- 2) what kind of algorithms will be applied;
- 3) what kind of data will be dealt with;
- 4) what kind of machine will implement the work.

There are many ways of separating the subtasks. Mainly there are four levels of parallelism (Hockney and Jesshope 1988, Crichlow 1988):

- 1) job level:
  - a) between jobs;
  - b) between phases of job;
- 2) program level:
  - a) between parts of program;
  - b) between different parts of data;
- 3) instruction level:
  - a) between phases of instruction execution;
- 4) arithmetic and bit level:
  - a) bit-serial;
  - b) bit-parallel.

At the highest level, the aim of a computer installation is to maximise the rate of processing of jobs. In the simplest analysis, each job may be considered as being divided into several sequential phases, each of which requires a different system program and system resources; the main requirement of computer architecture in allowing parallelism at job level is to provide a correctly balanced set of replicated resources. For example, any large computer installation provides several I/O channels or peripheral processors which can perform I/O in parallel with program execution, and provides a battery of disc and tape drives.

Within a program, there may be sections of code that are independent of each other and could be executed in parallel on



different processors in a multiprocessor environment. Some sections of independent code can be recognized from a logical analysis of the source code, but others will be data-dependent and therefore not known until the program is executed. In another case, executions on different parts of a data array may be independent of each other, even though different routes are taken through conditional statements contained in the code. In this case, each microprocessor can be given the full code, and as many parts of the data array as there are microprocessors can be passed in parallel through the code. These alternative methods of parallelism may be called program-structure parallelism and data-structure parallelism respectively.

Instruction parallelism is a lower level of parallelism; the processing of any instruction may be divided into several suboperations, and pipelining may be used to overlap the different suboperations on different instructions. Some FORTRAN compilers have been produced which can replace a DO-loop by one or several vector or array instructions and can therefore be executed much more effectively by machine instructions on suitable machines. The architectural features needed are then either pipelined vector computers (e.g. CYBER 205 and CRAY-1) or replicated processing elements in the case of the array processor (e.g. ICL DAP and BSP, Hockney and Jesshope 1988).

At the lowest level, ordinary CPUs process bits in the bit-parallel mode, i.e. they process all the bits of a number simultaneously; for example, they use 32 bits in 80386 PCs; 32 bits in the T800 transputer; and 32 bits in the ICL 3980 mainframe. However, special hardware is also used. Arithmetic processors have been designed either to perform the arithmetic in a bit-serial mode or in a bit-parallel mode. Both these types of processors can be arranged in

parallel. If for example, the memory is regarded as a two-dimensional array of bits holding one set of numbers with all the bits for one number stored in one row, then the bit-serial mode reads vertical slices of bits and processes all the first bits of the set of numbers simultaneously, then all the second bits, and so on. Such are the ICL DAP and STARAN machines. Conversely, the bit-parallel mode reads a horizontal slice of bits, *i.e.* all the bits for the first number: such are the CDC 7600 and CRAY-1 machines. For array processing, the new DAP 500 and DAP 600 have, respectively, 1024 Processor Elements (PEs) and 4096 PEs arrayed in a square array of  $32 \times 32$  or  $64 \times 64$ , each with a one-bit wide memory below it, thus giving a cuboid of memory Fig. 6.1. Memory addresses are common to all PEs (processor elements), so in a given instruction, each PE accesses a bit of memory at the same memory address; that is to say a complete 'plane' of memory is being accessed at the same time. Generally, each data item is held in the memory of a particular processor, with successive bits of each data item occupying successively addressed locations. Thus, a matrix of 16 bit values, for example, occupies 16 consecutive bit-planes of the memory. Some instructions provide access to one row of memory (corresponding to one row of PEs), or to one word of memory (32 consecutive bits in a row), selected from a memory plane (Hunt 1989).

### 6.3 ARCHITECTURE OF PARALLEL SYSTEM

The style of parallelism is very much dependent on the architecture of the parallel machine which is used for the work. A wide variety of different parallel architectures exists. It is very difficult to classify these designs. Some well established architectures, particularly the highly successful pipelined computers, may not fit into any simple class, or may fit equally well into

several classes. Hockney and Jesshope gave classification as below (1988):

1) SISD — single instruction stream/single data stream.

This is the conventional serial computer in which there is one stream of instructions (and therefore, in practice, only one instruction processing unit) and each arithmetical instruction initiates one arithmetical operation, leading to a single data stream of logically related arguments and results. It is irrelevant whether pipelining is used to speed up the processing of instructions or the arithmetic. It is what have previously been called a serial scalar computer. Examples are: CDC 6600 (unpipelined); CDC 7600 (pipelined arithmetic); AMDAHL 470V/6 (pipelined instruction processing).

2) SIMD — single instruction stream/multiple data stream.

This is a computer that retains a single stream of instructions but has vector instructions that initiate many operations. Each element of the vector is regarded as a member of a separate data stream, hence, excepting the degenerate case of vectors of length one, there are multiple data streams. This classification therefore includes all machines with vector instructions. Again it is irrelevant whether the capability of vector processing is realised by pipelining or by building arrays of processors. Examples: CRAY-1 (pipelined vector computer); ILLIAC IV (processor array); ICL DAP (processor array); OMEN-64 (processor array).

3) MISD — multiple instruction stream/single data stream.

This class includes specialised streaming organizations using multiple instruction streams on a single sequence of data.

#### 4) MIMD — multiple instruction stream/multiple data stream.

Multiple instruction streams imply the existence of several instruction processing units and therefore necessarily several data streams. This class therefore includes all forms of multiprocessor configurations, from linked main-frame computers to large arrays of microprocessors.

Over all these different parallel computers, there are two main techniques for introducing parallelism into computer hardware: replication and pipelining (Hockney and Jesshope 1988). Pipelining can be considered as replication which has been made possible through sequence, as each component of replication in a pipeline follows another in time. The difference between pipelining and array replication is that the parallel component operations of a pipeline are quite likely to perform different tasks, which when performed in sequence make up the operation required. Fig.6.2 illustrates the different ways of performing an arithmetic operation on serial, pipelined and replication array architectures. As an example, solving the problem of adding two floating-point vectors  $x_i$  and  $y_i$  ( $i=1,2,\dots,n$ ) to obtain the sum vector  $z_i = x_i + y_i$  ( $i=1,2,\dots,n$ ). The operation of adding any pair of the above elements may be divided into four sub-operations: compare exponents; shift; add; normalise.

Pipelined operations are performed by overlapping their simpler component operations using parallel hardware. This is performed such that at any given time, component parts of a sequence of operations are being processed in the pipeline. In this way, a single operation will share the pipeline with a number of other operations as it progresses through the various stages. Pipelining is an attractive form of parallelism, because pipelining does not create the same

communications problems found using array replication. A pipeline is designed to reflect the natural data flow of the operation being performed. After more than thirty years development, there are highly successful pipelined supercomputers: such as CRAY X-MP and CRAY-2; CDC CYBER 205 and ETA. These super computers are pipeline systems with shared memory, Fig. 6.3a. The transputer formed pipeline system is a pipeline system with ~~has~~ distributed memory, Fig 6.3b.

Although it is attractive, pipelining alone will not achieve the goal of unlimited computing power, which has motivated the use of parallelism in computer systems. This will only come about through the use of array replication, or indeed the use of both pipelining and array replication (Hockney and Jesshope 1988). The array replication will utilise either a fixed network (Fig. 6.4a) or a programmable connection network to reflect data flow (Fig. 6.4b). In the first case, the network may not necessarily reflect the data flow required in the operations being performed, because each processor has its own memory. Therefore, communication may be needed between the processors in some cases, but it is usually very minor (For an example of the author's system see Section 7.3.3). In the latter case, Fig. 6.4b, although programmable connection is more general, the costs are large, because the design is focused on how to switch data between processors. Typical array supercomputers are the ICL DAP; the Burroughs BSP; the Denelcor HEP; and the latest transputer systems.

#### 6.4 COMMUNICATION WITHIN PARALLEL SYSTEM

In parallel systems, a small number of processors may communicate efficiently using bussed systems or shared memory. For a larger number of processors, the inherent sequentiality of these communication

methods produces bottlenecks. The designer is then forced to consider systems in which data is switched between processors. This switch can take place either in a fixed network, in which distances grow with the number of processors, or in a programmable connection network, in which the costs follow a square law. The theory and construction of switch networks are fundamental to the success of large-scale parallelism, which has now become feasible through the exploitation of Very Large Scale Integration (VLSI) technology. Large scale replication is not viable unless connections can be established either between processors or between processors and memory in a programmable manner.

The communication can be made in different ways:

- 1) bus on the same chip between processor and local memory;
- 2) permanently wired;
- 3) human controllable hardware links;
- 4) programmable link switch under electronic control, e.g. the INMOS C004 chip;
- 5) automatic configuration electronic switch, e.g. the Meiko;
- 6) dynamic reconfigurable electronic switch, for example the forthcoming system T9000

## 6.5 PARALLEL OPERATING SYSTEM

After discussing the level of parallelism used in splitting a task and the ways in which the machine will be implemented, the next stage is to configure these subtasks onto the multiprocessors. Parallel operating systems are designed specifically to exercise the parallel hardware features, and to control the execution of parallel processing software. The parallel strands within the program have to

be identified; they must be allocated to available hardware units; and their interactions ought to be adequately controlled. Several kinds of parallel operating system are discussed below:

1) Master-slave organization

Only one processor, the master, executes the operating system, and the other processors operate in a slave mode. The Transputer Development System, TDS, which the author used in this work belongs to this kind of parallel operating system.

2) Floating supervisor control

All the processors execute the operating system, and the supervisor role floats from processor to processor.

3) The distributed operating system

One operating system running in a MIMD machine. Each processor-memory node contains a separate copy of the kernel of the operating system and they are connected by a harness.

## 6.6 PARALLEL LANGUAGE

A parallel language must control the structure of the program and provide data to the various parts of it to express the inherent parallelism of the problem being tackled. These languages allow the specification of different parallel streams of activity within a program. They also provide mechanisms for interaction among these parallel task.

There are several parallel languages available for writing programs, such as Occam, ADA, parallel C, parallel PASCAL, and parallel FORTRAN. All these compilers are now becoming widely available for transputers. Occam is the language which the author is using in this work.

## 6.7 APPLICATIONS OF TRANSPUTERS

Although many of the principles of parallelism discussed here had been implemented in the past either on specialised hardware or on expensive supercomputers such as the Cray, the recent development of transputers has provided a cheap, efficient, and flexible means of bringing parallelism to the ordinary user.

The applications of transputers have developed widely and very rapidly. They widely cover Image Processing, Signal Processing, Communication, Medical 3-D and 2-D, Remote Sensing, Robotics, Real-Time Control, Simulation, Object Recognition and Feature Detection, Numerical Techniques, Data processing, etc. Image processing deals with huge amount of data (up to 1024\*1024 pixel or even more). Each pixel sometimes requires hundreds of operations. Real



time video control needs even faster machines. The transputer provides an efficient and powerful system for image processing. The author's implementation of a multi-processor transputer system is described in the next Chapter.

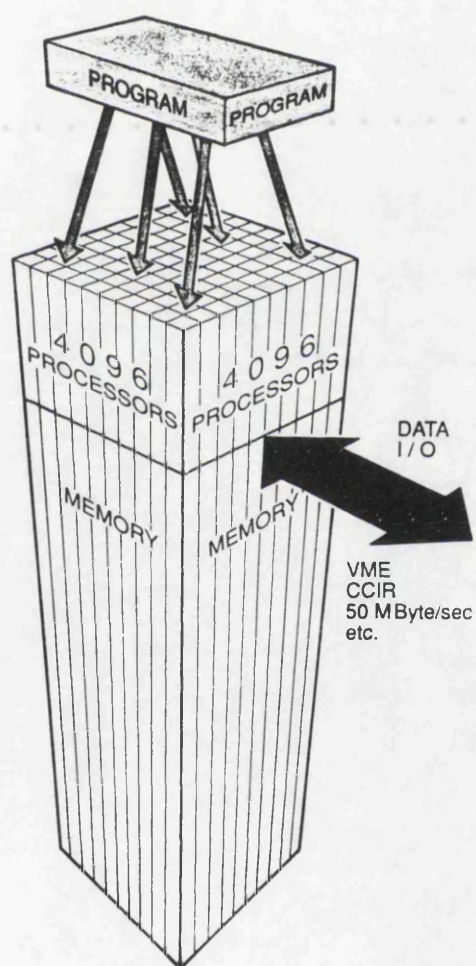


Fig. 6.1 The architecture of DAP 610 (Hunt 1989).

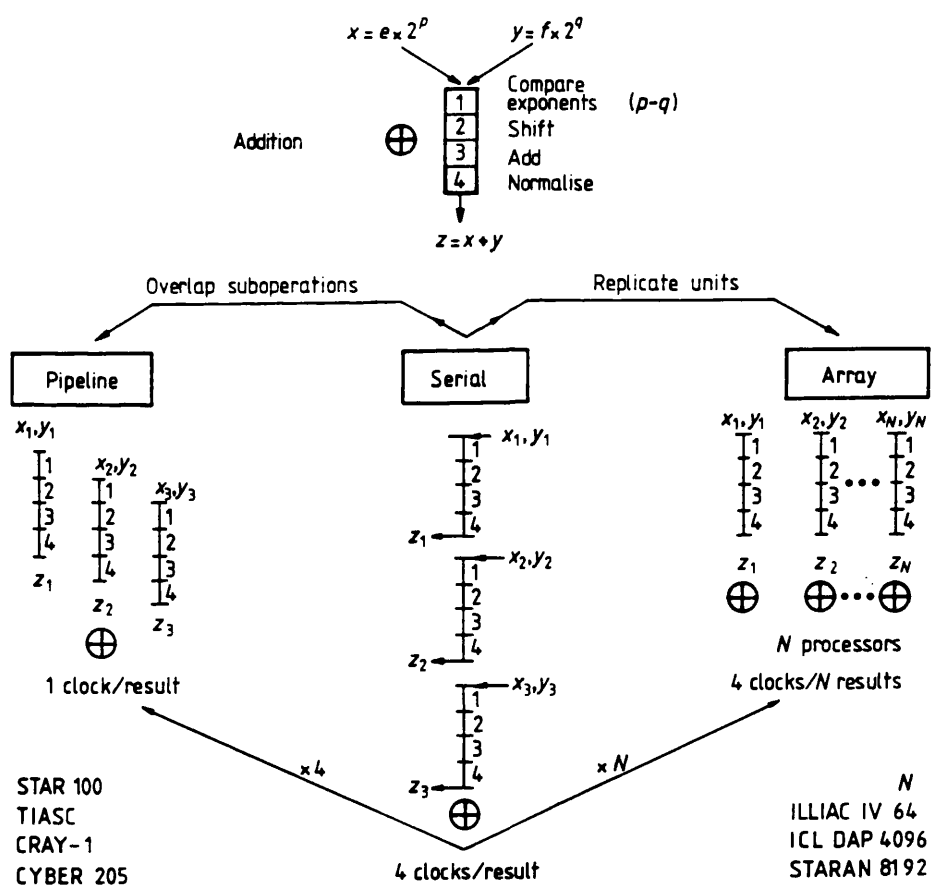


Fig. 6.2 Comparison of serial, pipelined and array structures (Hockney and Jesshope 1988).

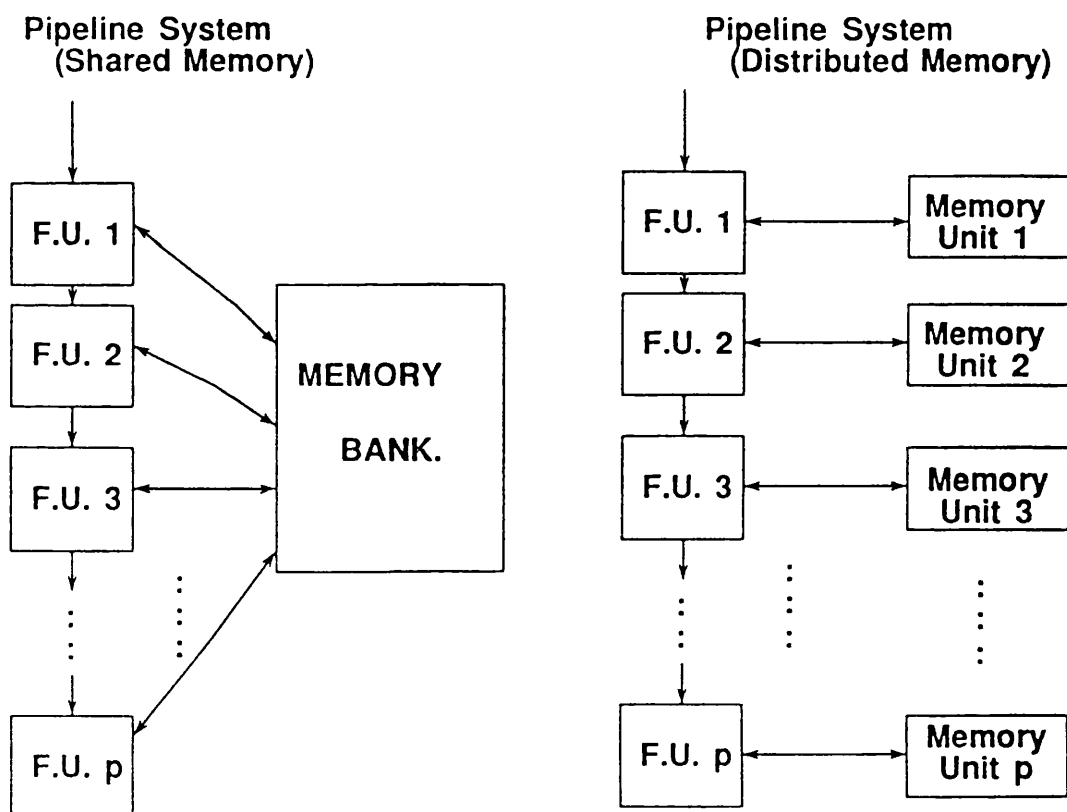


Fig. 6.3 Structure of pipeline processors (Soraghan 1989).  
(a) with shared memory.  
(b) each processor with its own local memory.

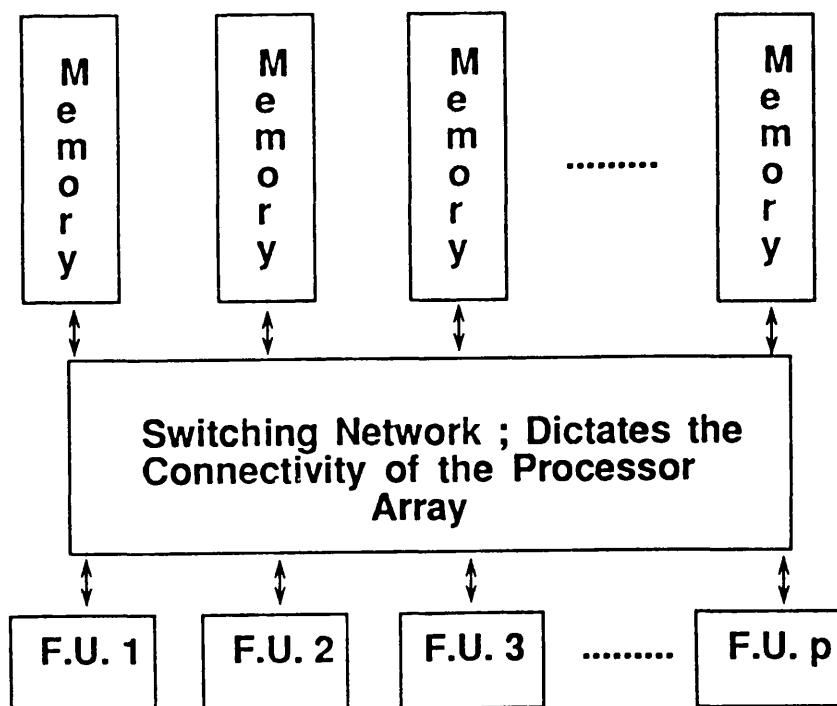
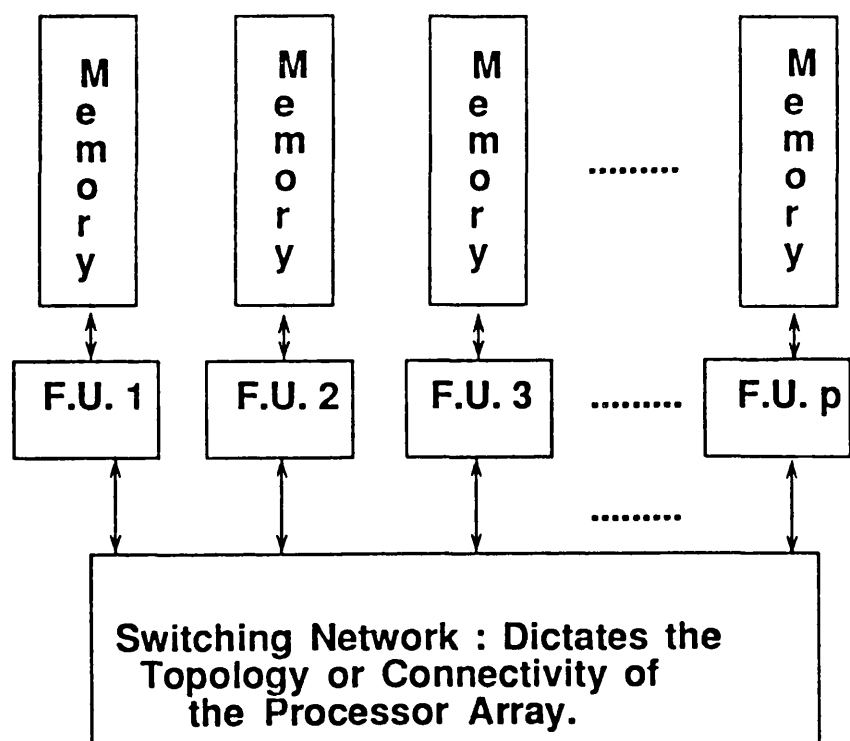


Fig. 6.4 Structure of array processors (Soraghan 1989).

- (a) SIMD mode with fixed connections between processors and their local memory
- (b) MIMD mode with programmable connections between processors and memory.

## CHAPTER 7 PARALLEL PROCESSING — PRACTICE

### 7.1 TRANSPUTER, TDS, AND OCCAM

The transputer is a microcomputer on a single chip containing processor, memory, and links which provide communication connections with other transputers. There are three main series of transputers available: T2; T4; and T8; and an improved T9 series is expected soon.

The T2 series of transputers are 16 bit 10 Million Instructions Per Second (MIPS) processors with 2 KByte high speed on-chip Random Access Memory (RAM), non-multiplexed 20 MByte/sec memory interface, and 4 pairs of 20 MByte/sec inter-transputer links. (The exact performance depends on the clock speed, which is being increased progressively). The T4 series are 32 bit 10 MIPS processors, with 2 KByte high speed on-chip RAM, 32 bit wide 25 MByte/sec memory interface, 4 GByte linear address space, and 4 pairs of 20 MByte/sec inter-transputer links. The T8 series are 32 bit 10 MIPS processors, with integral 64 bit Floating Point Unit, 4 KByte high speed on-chip RAM, sustained performance of 1.5 MFlop/sec, 32 bit wide 25 MByte/sec memory interface, configurable on-chip memory controller, and 4 pairs of 20 MByte/sec inter-transputer links (INMOS 1987). Commercial production of the new INMOS T9000 is expected very soon. It has a 32 bit integer processor, a 64 bit floating point unit, 16 KBytes of cache memory, a communications processor and four high bandwidth

serial communications links on a single T9000 chip. T9000 transputers are expected to be 5 - 10 times faster than T800 transputers are today. All kinds of available transputer chips are listed in Table 7.1. Apart from the early work, which used T414 transputers, T800 (Fig. 7.1) transputers were used in author's research (INMOS 1988c).

What distinguishes the INMOS transputer from its competitors is that it has been designed to exploit Very Large Scale Integration. Like some other recent microprocessors, it has a Reduced Instruction Set Computing architecture. The area of silicon that has been saved has been used for on-chip RAM and an off-chip external memory interface (see Fig. 7.1). The external communications system through the links is a direct implementation of the point-to-point asynchronous communications required by Occam processes. Thus, transputers can be linked together to form multiprocessor networks in many ways: the basic structures can be pipeline, tree, and array structures, see Fig. 7.2. As many transputers as needed can be connected together to form a multiprocessor network with a mixture of these basic structures within it. In addition, several processes can be run in parallel within a single transputer, but at present this is done by time-slicing. Another important aspect is that the total memory used can be increased as more transputers are incorporated in the network.

The reasons for choosing transputers rather than other computing machines to implement the present work were:

- 1) Transputers are quite efficient processors; 32 bit integer arithmetic for the T4 series, 32 bit floating arithmetic for the T8 series.

- 2) Transputers have large local memory: the boards installed in the PC for this research had 2 MByte memory for each transputer; the University's MEIKO machine has 4 MByte memory for each transputer; and the latest Parateck's Super-memory Transputer Module (TRAM) PST204 design has 72 MByte local Dynamic RAM (DRAM). The communication of the transputer with its local memory is much quicker than communication with global memory would be for the same number of processors.
- 3) A parallel system of linked transputers is flexible. The system can have pipeline, array, tree structures; or any combination of these as required by the research.
- 4) A parallel system consisting of transputers is extendable. With 4 links on each transputer, as many as are in hand can be connected together to do more jobs with program-structure parallelism or to analyse larger images with data-structure parallelism (Section 6.2).
- 5) Many parallel languages can be used on transputers. The compilers are available, and there is no serious programming problem and a reasonable degree of portability.
- 6) Transputers are relatively cheap, typically about \$55 for T400's, \$108 for T425's, and \$180 for T800's (all 20 Mhz). The system can thus be developed step-by-step according to the economic situation.
- 7) A transputer system can incorporate other processors; in particular, Digital Signal Processors and Vector Processors can be inserted into the system if necessary.

Occam is a parallel language which treats concurrence in a simple, consistent, and formal manner. Occam was developed closely in conjunction with the transputer, so that the central language features



have built-in hardware support. The language enables a transputer network system to be described as a collection of concurrent processes communicating with each other through channels. Once it had been decided to use transputers, it was decided to use Occam in the author's research, because Occam and the transputer had been designed together, and it was anticipated that programs written in Occam would run slightly more quickly than programs written in other languages, since the software structure of the language and the hardware structure of the processor were closely matched.

Occam programs can be described as a combination of processes. There are three primitive processes:

- input -- receives a value from a channel,
- assignment -- changes the value of a variable,
- output -- sends a value to a channel.

Processes are combined to form sequential, parallel or alternative constructs:

- SEQ -- the component processes are executed one after another,
- PAR -- the component processes are executed concurrently,
- ALT -- whichever component process is ready to communicate first is executed first.

The Transputer Development System (TDS) is an integrated programming environment developed by INMOS to support the programming of transputer networks in Occam. It comprises an integrated editor, file manager, compiler, and debugging system. It configures the subtasks onto the multiprocessors. It is, however, sometimes difficult to use, and consideration is being given to using the alternative

'Occam Toolset' for future work.

## 7.2 HARDWARE SYSTEMS

During the last two years of using parallelism, four versions of the system were developed for the work. At the same time, the original algorithm was improved, and execution time decreased. Table 7.2 shows the Top-contouring running times on different systems.

The Top-contouring algorithm beginning was an inefficient PC Fortran version (Table 7.2). It needed 4 hours to run on a NIMBUS 80386 PC and about half an hour to run on a mainframe to analyse one single image. Although the use of a single transputer gave some improvement, to speed up the calculation and apply it to a whole set of images, parallel processing was chosen for the work to solve the problem.

In preparation for the parallel system, a single T414 transputer board, IMS B004 IBM PC Add-in Board, was installed inside the NIMBUS, on loan from SERC. It enables users to evaluate and demonstrate the use of transputers. The board is one of a family of compatible evaluation boards containing a 32 bit transputer and standard buffered INMOS link connections. The board provides a powerful upgrade to the IBM PC XT or AT. A multi-transputer network can be built up by linking several boards together. Such a network is easily constructed by supplying power to each board from the PC bus, and connecting at least one link on each board to another board in the network using the cables supplied (INMOS 1987). Before the board can be used with the TDS and a PC, link and reset connections need to be made between the

transputer and the PC bus by plugging in a Link Jumper and a Reset Jumper on the edge of the board.

In the first multi-transputer system used for this work Fig. 7.3, a four-transputer board, IMS B003 Transputer Evaluation Board, was connected as a subsystem to the previous B004 single transputer board, which was used as the master transputer (see Fig. 7.3). The subsystem connection allows the lower B003 board to be controlled by the B004 board (when the B004 board is reset, the B003 board is reset at the same time). The B003 board enables users to evaluate and demonstrate the use of multiple transputers. The four 32 bit transputers on this evaluation board make it a very powerful processing tool capable of up to 40 MIPS. Each transputers has its own 256 KByte dynamic RAM. The four transputers are permanently hard-wired in an array using link pairs 2 and 3 (see Fig. 7.3). The other link pairs, 0 and 1, are brought to the edge of the board, so that the user can either cross-connect them or connect them to other boards.

On the introduction of T800 transputers to this work, the whole system was upgraded, Fig. 7.4. The master board was replaced by a Transtech TSB04 single transputer board with one T800 rather than the T414. The subsystem was replaced by an IMS B008 Motherboard (Fig. 7.5) with 4 IMS B404 TRAMs (Fig. 7.6) plugged into it. Each TRAM has one IMS T800 transputer, 32 KByte of zero wait-state SRAM, 2 MByte of single wait-state DRAM, subsystem controller circuitry, and communications via 4 INMOS serial link pairs. The TRAM has 16 active pins connecting it to the motherboard slots. Link pairs 1 and 2 from each of the TRAM slots are hard wired on the IMS B008, such that the TRAMs, when plugged in, form a pipeline of processing elements. (An

optional break in the middle of this pipeline was not used here.) The remaining links can be 'softwired' using an INMOS IMS C004, programmable link switch, incorporated on the IMS B008. By using the C004, an array structure was set up as a secondary communication system superimposed on the pipeline. The IMS C004 device is controlled by an IMS T212 16-bit transputer. To control the motherboard, the link pair 1 of the T212 transputer must be connected with the remaining link of the master transputer (link 2). For further details of setting up the motherboard see INMOS (1988.d, and 1989.b). Additionally, a Transtech Harlequin T800 driven graphics board (Transtech 1988) was used at the end of the pipeline for displaying pictures on a secondary monitor (Fig. 7.7). The six-transputer network shown in Fig. 7.4, 7.8 and 7.9 formed the basic PC version parallel system for this research, (Appendix A).

At the end of the SERC loan, the work was moved to the Glasgow University's Meiko computer machine. The Meiko computer originally had 32 T800 transputers which were semi-permanently arranged by the computer manager into groups called domains. One of these domains, called T6G, was based on the NIMBUS system. The system of T6G is shown in Fig. 7.10. This is not divided into separate boards as in the PC, but dotted lines have been drawn across the diagram to emphasize the functional blocks into which the system was divided. All the links (except the pair for the host) are connected electronically. The diagram shows the wiring used most often, and it will be seen that this is identical to that used in Fig. 7.4, except for the numbering of the links, and for the continuation of the pipe back to the top transputer. The MK52 transputer controls a second monitor on which images are displayed. Each of the six transputers has 4 MByte memory.

The T6G domain was usually run from a Sun Workstation with the second monitor beside it; but it could be run from a terminal attached to the University's computing network if it was not necessary to see the images. After the main research had been completed, the Meiko was improved, so that users could set up their own domains when logging on.

### 7.3 SOFTWARE SYSTEMS

#### 7.3.1 Introduction

In the author's image analysis system there are three main stages: obtain the image; analyse; and display the result. These three stages, in parallel processing terms, are three subtasks, forming the basic pipeline structure of the parallel practice, as shown in Fig. 7.11. These three subtasks are located on three multiprocessor blocks, which can be seen more clearly in Fig. 7.4, where each PC add-in board runs one subtask. Almost all analysis in the T6G six transputer domain in the Meiko machine used the same parallel structure, see Fig. 7.10. (Corresponding to different calculation requirements in each stage, The author could design different multiprocessor blocks and link them in the pipeline, each block containing one transputer or more as required).

The top block, in Fig. 7.4 and 7.11 obtains the image, contains a single transputer which specifically deals with:

- 1) the interface with the host, which includes input and output of pictures and other data and messages from the host;
- 2) sending the original picture and messages down to the working block

and to the display block, and getting results and messages from these blocks;

3) summarizing the results which come out of the working block.

In the analysis multiprocessor block, data-structure parallelism (see Section 6.2) has been chosen to implement the new algorithms developed in this research, because:

- 1) It is much easier to write the program with every working processor executing almost the same code on different portions of data than with each processor executing different calculations on the same data.
- 2) For image analysis calculations, which inevitably involve large amounts of data calculation, data-structure parallelism often needs less time for data transfer than process parallelism.
- 3) Several of the new algorithms use large filters to scan over the image, so in practice each transputer must receive all of its data before it can begin work.
- 4) When using data-structure parallelism, it is easy to extend the program to include additional stages of programing.
- 5) Data-structure parallelism places equal loads on the working transputers, whereas program-structure parallelism would load them unequally, so some of the working transputers would then be idle for some of the time.

The lowest block is a graphics board, which is set up to display any image which is sent to it.

### 7.3.2 Data-structure Parallelism

Currently, each image data array has been divided into four quarters or tiles (Fig. 7.13). Each quarter of the data array was mapped onto one of the four transputers at the corresponding position in the analysis multiprocessor block (Fig. 7.15). Two problems which then arose are discussed in the next two paragraphs.

Before deciding to divide the image into tiles, consideration was given to dividing it into horizontal strips (Fig. 7.13). Some of the algorithms use large filters of 20-30 pixels radius for the later stages of the calculation. Before using these filters, the results from the previous stages are 'swapped' along the internal borders between the working transputers. (Later, the author will show how to eliminate swapping in 'production' versions of the code; but the use of swapping makes it easier to write code when testing new algorithms.)

Many image analysers leave unprocessed borders around the edges of the image, the width being equal to the radius of the filter. When large filters are used, the loss of this unprocessed area is significant. For some of the algorithms used here, it was found possible to process right up to the edges of the image even where very large filters were used. To do this, an area along the image edges with width equal to the filter radius is reflected outwards. (This method is not suitable for some algorithms.)

In the PC system, which had only 2 MByte per transputer, it was sometimes necessary to place successive versions of the same picture

in the same workspace. This involved offsetting one version from the centre as explained by Smart and Leng (1990); but it is not explained in detail here, because it is better to use enough RAM. A method of estimating how much RAM is needed is given in Appendix B. In the Meiko system, the workspaces on each working transputer are initialized as  $512 \times 512$ . At present, the images are  $512 \times 512$  pixels, each working transputer has one quarter of the image, which is  $256 \times 256$ , mapped in the centre of the workspaces leaving 128 pixel wide borders. However, the width of swapping and filling edges can be of any value up to 128 pixels equally or unequally. Fig 7.12 shows the arrangement of the workspace located on the top left working transputer.

Fig. 7.13 illustrates swapping and filling edges for tiles and strips respectively, for four transputers analysing a  $512 \times 512$  image. The width of edge swapping and filling should be equal to (or larger than) the radius of the filter. Table 7.3. compares the amounts of the swapping and filling for a width of 20 pixels. In this case, tiles involve less swapping and filling than strips; but the pixel-by-pixel calculation overestimates the difference, because the movements are by block transfer.

### 7.3.3 Communication

Several processes can run in parallel on a single transputer. Because this is done by time slicing, there are three processes running in parallel on each working transputer in the analysis multiprocessor block (see Fig. 7.14). They are:

- 1) The 'Tee Process' is a junction which receives data and messages from the upward transputer along the pipeline and passes these both



to the 'Engine Process' on the same transputer and to the downward transputer along the pipeline. Alternatively, it receives results from the Engine Process and passes these to the downward transputer. All four Tee Process junctions link together to ensure a positive direction of transfer from pipehead to pipetail through the multiprocessor block without impediment.

- 2) The 'Resp Process' alternatively receives messages and results from the downward pipeline or from the Engine Process on the same transputer and passes these upwards. All the four Resp Processes link together to ensure a negative direction of transfer from pipetail to pipehead through the multiprocessor block without impediment.
- 3) Each Engine Process implements arithmetic sequentially on one quarter of the image; but the four Engine Processes run simultaneously.

The main pipe formed by the Tee and Resp Processes was designed as a pair of uni-directional pipes, so that communications in the up- and down- directions could never clash and cause it to hang (see Fig. 7.15). Some details of the code are given in Appendix B. When debugging programs, the main pipe automatically passes commands and messages through the system when temporary code in the main processes requires these. (The Meiko computer has an additional debugging channel via the system control links; but this has not been used, because it would not be available on other systems, and the author's pipes were already in use.)

Inside the analysis multi-processor block, since the method of dividing the image demands, the four transputers are linked together

in an array structure by another set of links between them. These links are used for swapping, as shown in Fig. 7.15, and they allow very quick interconnection without tangling with the pipeline transferring between the different transputers. This arrangement of the links is optimised for four transputers working on four quarters of the image. If the image were divided into 16 tiles, and mapped onto 16 transputers, then some of the swaps would have to use the pipeline (see Section 7.4).

In the Meiko, there is an extra link pair connecting the Display Block at the end of the pipe to the Top Block. In this case, pipehead and pipetail of the system have been linked together to form a circle. This link pair has proved useful for retrieving results from the end of the pipeline in the version of the code which the author supplied to a research student in the same group.

#### 7.3.4 One Mapping Process

Control of a parallel system of computing must be considered carefully in order to keep all the parts of the system working in phase with one another. Fig. 7.16 gives an example of a flowchart which was drawn when designing the control structure of the Top-contouring mapping program (Chapter 3) on the six transputer network installed in the PC.

Fig. 7.17 illustrate the use of four working transputers to process four quarters of an image in parallel. The transputers are calculating the intensity gradient and encoding the angles (see Section 3.3.2) for a fuzzy edged circle. The original circle edge is

from green (high) through blue to black. The top left transputer had already finished encoding (horizontal: red; down-left: yellow; vertical: green; down-right: blue); the top right was encoding; bottom left was calculating the intensity gradient; bottom right had not started calculating the intensity gradient.

#### 7.4 FUTURE DEVELOPMENTS

Plans involving additional parallelism have been made both for software and for hardware to build up a more extensive system specially for the analysis of the microstructure of scanning electron micrographs.

Normally sets of about 24 or even more images were taken from each sample. So some of the programs have been extended to analyse a whole set of up to 99 images automatically. The programs could be extended again to analyse all the samples from one mechanical experiment.

Two multiprocessor blocks may have to be installed in the present system (Fig. 7.18).

- 1) After obtaining the image, a preprocessing block may be needed to deal with image rectification or smoothing to remove noise from the image ready for analysis.
- 2) After analysis, the resulting image can be classified according to the characteristics of different textures or other statistical measurements may be made.

In this case, several images will be passing down the pipeline one in each block, and processed simultaneously.

Some electron microscope laboratories are planning to increase the image size from 512\*512 to 1024\*1024 pixels. To follow this development, each multiprocessor block will need more transputers. For example, the analysis block could have sixteen transputers, and each transputer would then still have 256\*256 size tiles.

From the experience obtained analysing small images divided into quarters, several changes have been identified which may be helpful, especially when analysing large images in large sets. (1) Dividing the image into strips rather than tiles will simplify the intercommunications, particularly if more than 4 working transputers are used. (2) Some postprocessing routines are easier to write using strips than tiles. (3) If the sizes of the filters are all known in advance, then swapping can be eliminated by preloading, i.e. each working transputer starts with extra lines of data, so that the central strips on each transputer initially overlap their neighbours.

Table 7.1 The transputer product range

Family	Part number	Speed (MHz)	On-chip SRAM	Serial links	Package	Process
<b>T2 – 16 bit CPU</b>	T222–G17M	17.5	4K	4	68 PGA	Mil–Std–883C
	T225–G20S	20	4K	4	68 PGA	Commercial
	T225–G25S	25	4K	4	68 PGA	Commercial
	T225–J20S	20	4K	4	68 PLCC	Commercial
	T225–F20S	20	4K	4	100 CQFP	Commercial
<b>T4 – 32 bit CPU</b>	T400–G20S	20	2K	2	84 PGA	Commercial
	T400–J20S	20	2K	2	84 PLCC	Commercial
	T400–X20S	20	2K	2	100 PQFP	Commercial
	T425–G20S	20	4K	4	84 PGA	Commercial
	T425–G25S	25	4K	4	84 PGA	Commercial
	T425–J20S	20	4K	4	84 PLCC	Commercial
	T425–F20S	20	4K	4	100 CQFP	Commercial
<b>T8 – 32 bit CPU +64 bit FPU</b>	T800–G17M	17.5	4K	4	84 PGA	Mil–Std–883C
	T801–G20S	20	4K	4	100 PGA	Commercial
	T801–G25S	25	4K	4	100 PGA	Commercial
	T805–G20S	20	4K	4	84 PGA	Commercial
	T805–G25S	25	4K	4	84 PGA	Commercial
	T805–G30S	30	4K	4	84 PGA	Commercial
	T805–J20S	20	4K	4	84 PLCC	Commercial
	T805–F20S	20	4K	4	100 CQFP	Commercial
<b>T9 – 32 bit CPU +64 bit FPU</b>	T9000–F40S	40	16K	4	208 CQFP	Commercial
	T9000–F50S	50	16K	4	208 CQFP	Commercial
<b>T2/T4/T8 peripherals</b>	C011–P20S	Link adaptor to bus or I/O			28 PDIL	Commercial
	C011–E20S	Link adaptor to bus or I/O			28 SOJ	Commercial
	C011–S20M	Link adaptor to bus or I/O			28 CDIL	Mil–Std–883C
	C012–P20S	Link adaptor to bus			24 PDIL	Commercial
	C004–G20S	32 way crossbar switch			84 PGA	Commercial
	C004–G20M	32 way crossbar switch			84 PGA	Mil–Std–883C
<b>T9 peripherals</b>	C100–F10S	System protocol converter			100 CQFP	Commercial
	C104–F10S	Packet routing switch			208 CQFP	Commercial

### Further information

INMOS is a member of the SGS–THOMSON Microelectronics Group, and supplies high performance microprocessors, systems products and colour graphics devices worldwide. The company has sales offices throughout the world,

supported by a network of experienced Field Application Engineers to assist with the design—in of the T9000. For further details please contact your local SGS–THOMSON Microelectronics sales office.

Table 7.2 Time (minutes) for Top-contouring mapping and display, approx.

Method	Analysis	Display	Remarks
hand mapping	960	--	A4 prints
PC	200	3	simple code
mainframe	20	--	no display
single T414	2	3	display via host
6 * T800 in PC	0.6	0	includes display
6 * T800 Meiko	0.4	0	includes display

These are timings taken during development not benchmarks.

PC = 80386/7 16 MHz;

mainframe = ICL 2890, run time, 40 other users;

T414 = 20 MHz;

T800 = 20 MHz.

**Table 7.3** The calculations of swapping and filling for mapping 512\*512 image onto four transputers cutting into either tiles and strips with 20 pixels radius filter.

Operations	pixels
<b>Tiles</b>	
up tile swapping	256*20
down tile swapping	256*20
left tile swapping	(256+20)*20
right tile swapping	(256+20)*20
total inter-transputer swaps	21280
parallel	(256+20)*20
upwards filling	
downwards filling	
parallel	(256+2*20)*20
leftside filling	
rightside filling	
total intra-transputer filling	11440
<b>Strips</b>	
parallel	512*20
0-1 strip swapping	
2-3 strip swapping	
1-2 strip swapping	512*20
parallel	512*20
1-0 strip swapping	
3-2 strip swapping	
2-1 strip swapping	512*20
total inter-transputer swapping	40960
parallel	512*20
0 strip upwards filling	
3 strip downwards filling	
parallel	(128+2*20)*20
0 strip left filling	
1 strip left filling	
2 strip left filling	
3 strip left filling	
parallel	(128+2*20)*20
0 strip right filling	
1 strip right filling	
2 strip right filling	
3 strip right filling	
total intra-transputer filling	16960

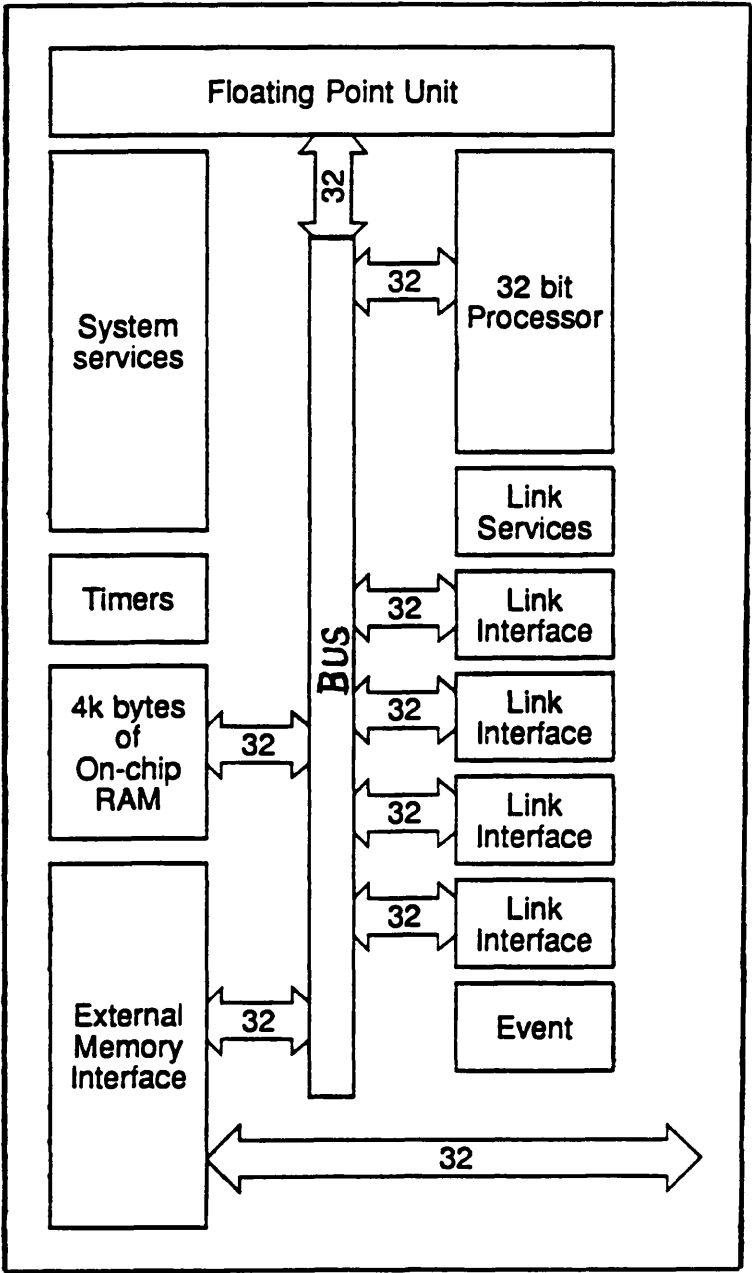


Fig. 7.1 IMS T800 block diagram (INMOS 1988a).



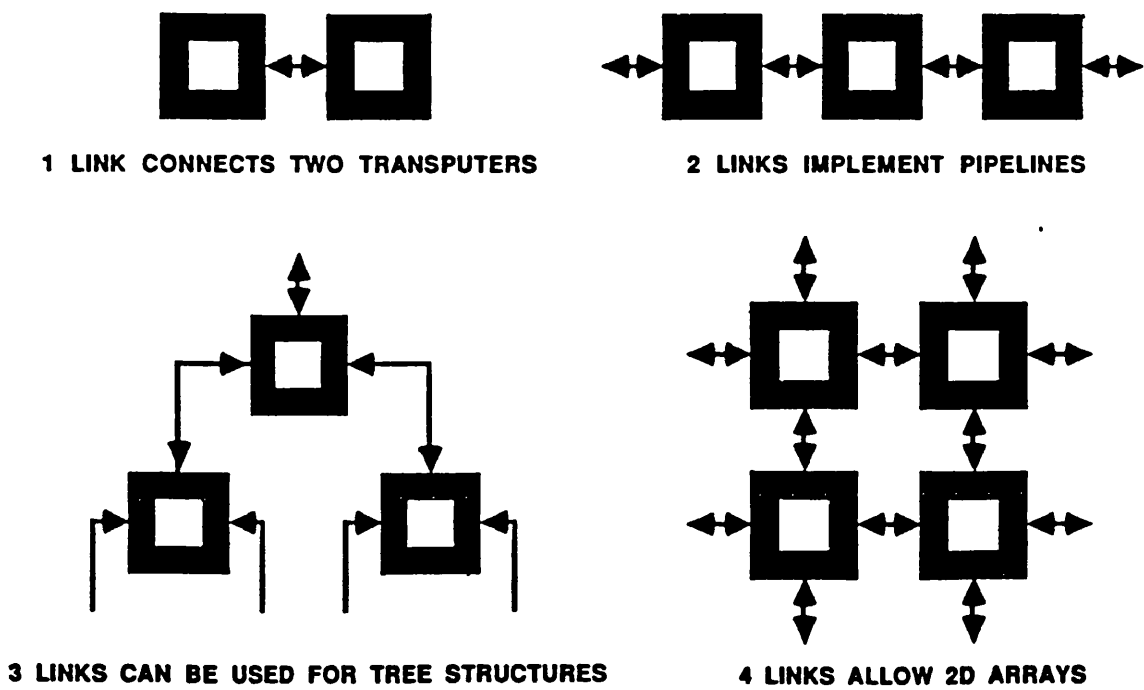


Fig. 7.2 Basic structures formed by linking transputers together (INMOS 1988a).

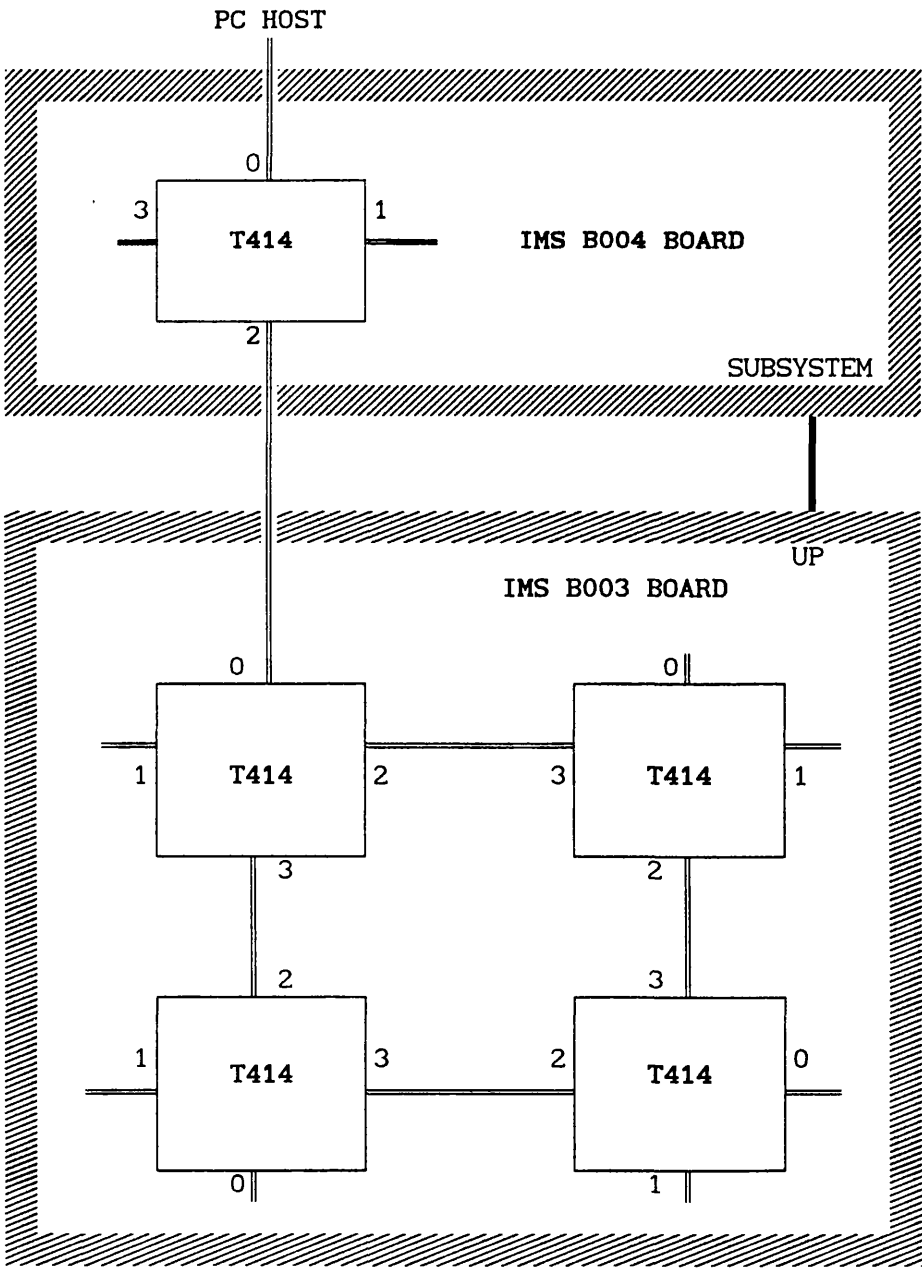


Fig. 7.3 Wiring of a system of 5\*T414 transputers on 2 boards in a RM Nimbus PC.

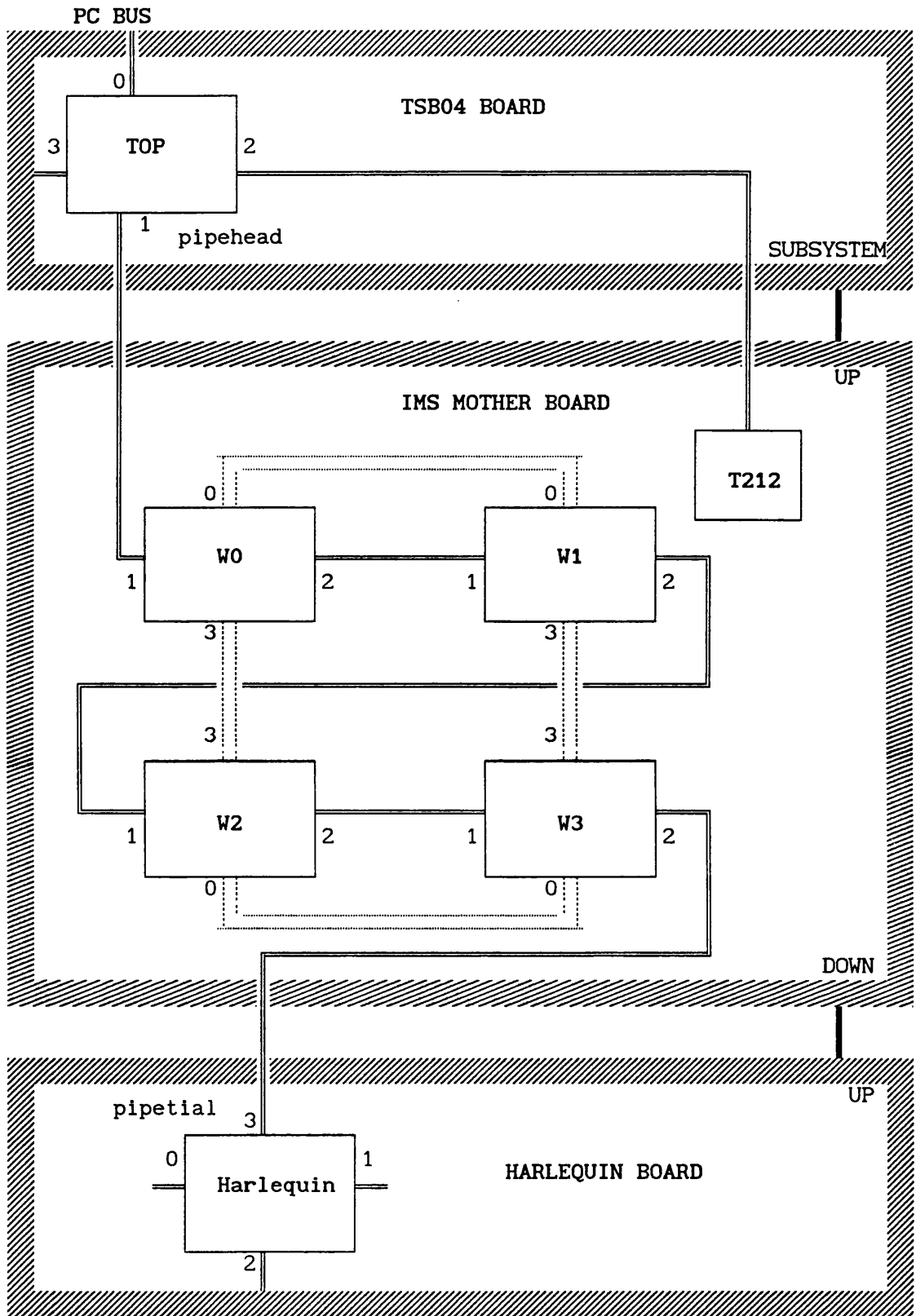


Fig. 7.4 Wiring of a system of 6\*T800 transputers on 3 boards (Transtech TSB04, IMS B008 Motherboard, and Transtech Harlequin) in a Tandon PC. (The T212 transputer controls the mother board)

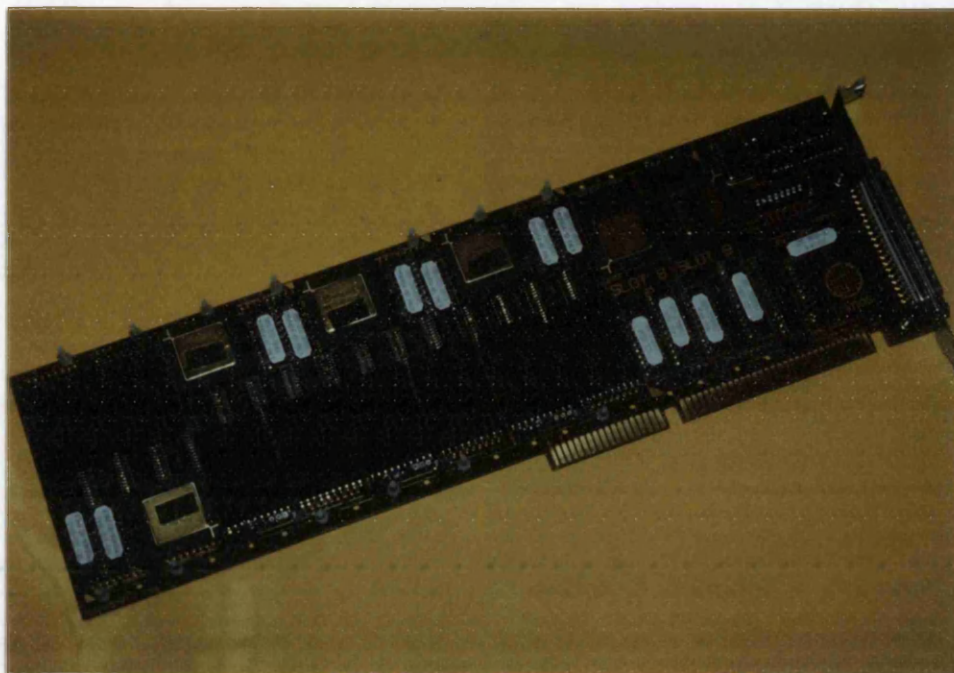


Fig. 7.5 INMOS Mother Board with 4 T800 TRAMs.

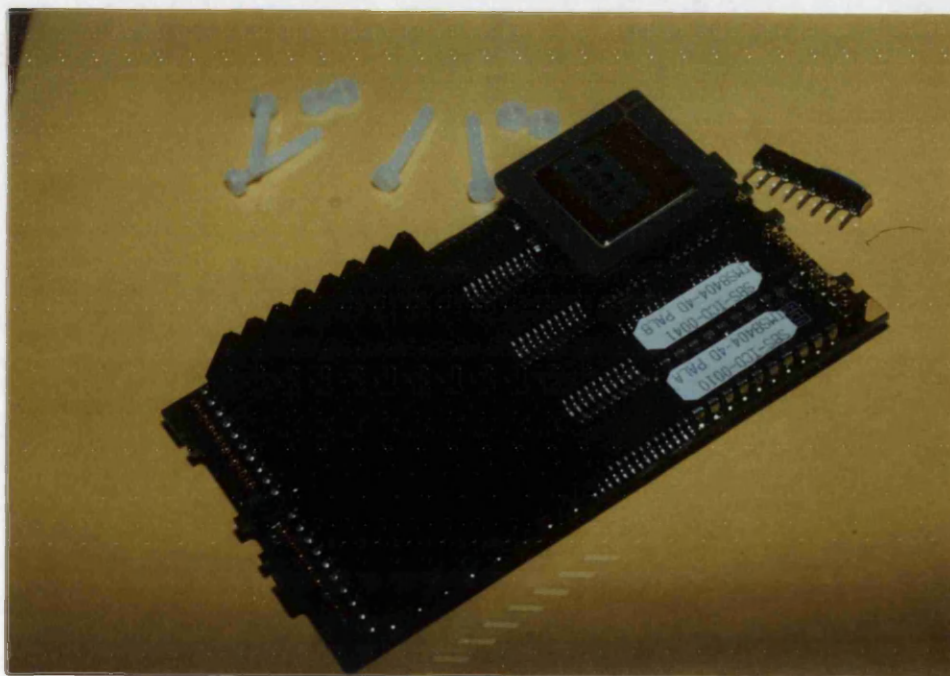


Fig. 7.6 T800 TRAM with 2 MByte memory.



Fig. 7.7 The PC showing the second monitor.

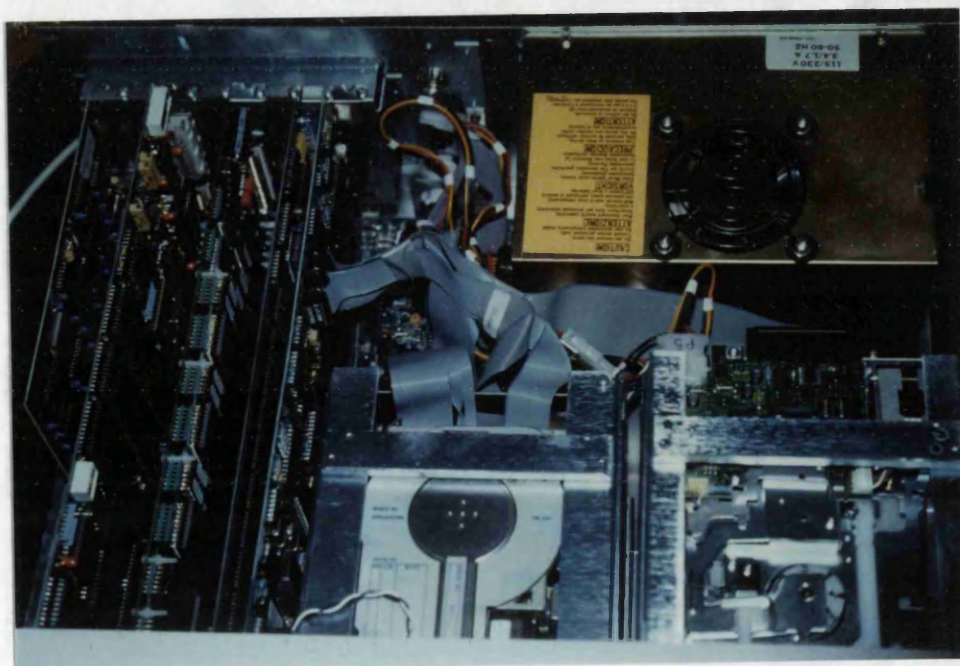


Fig. 7.8 Overview interior of author's PC based multi-transputer system.



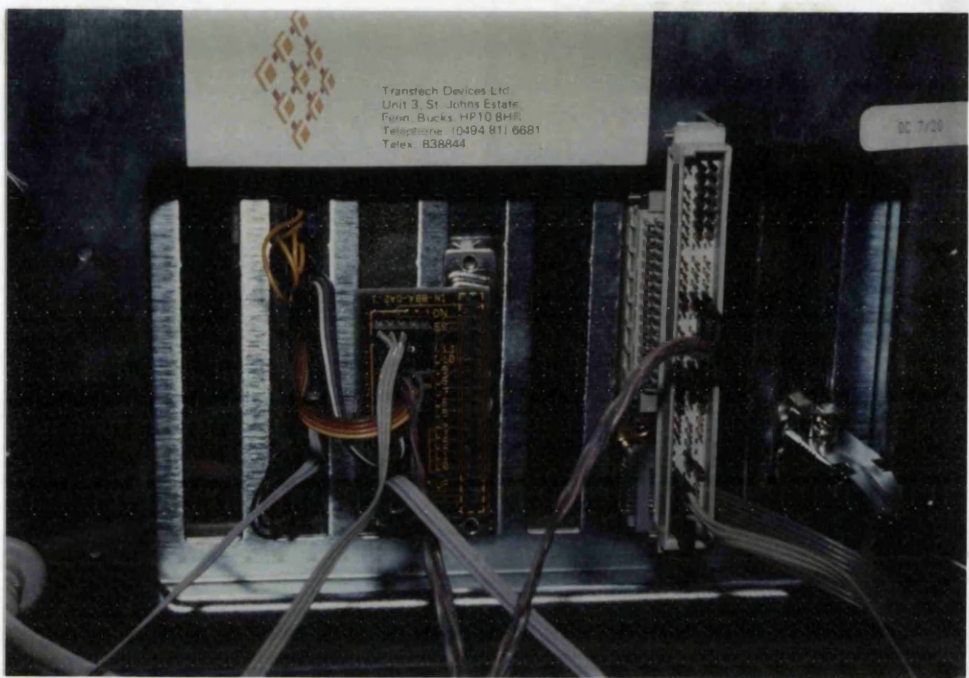


Fig. 7.9 The links of the author's multi-transputer system.

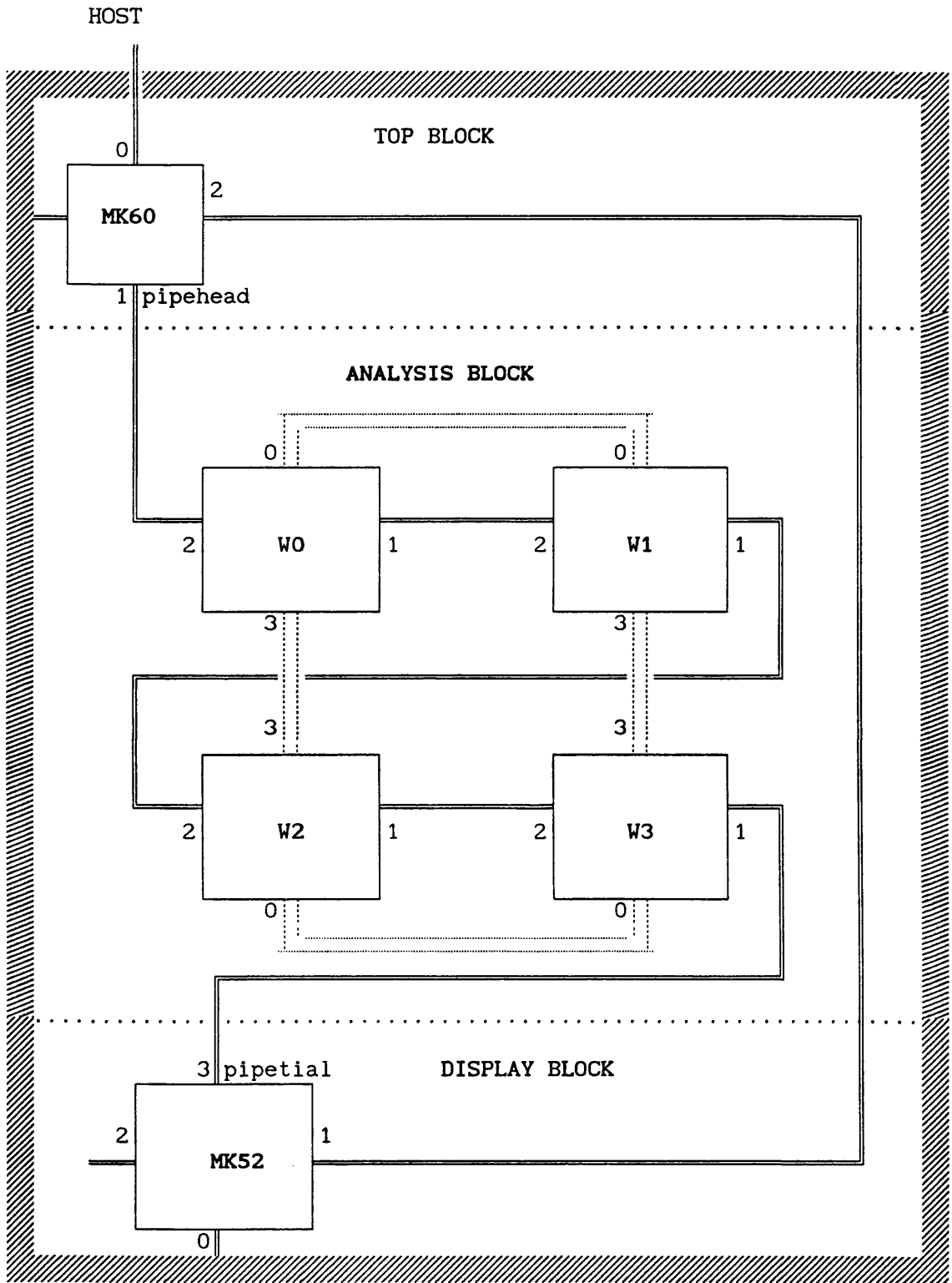


Fig. 7.10 Wiring of a system of 6\*T800 transputers (T6G domain) in Meiko computer.

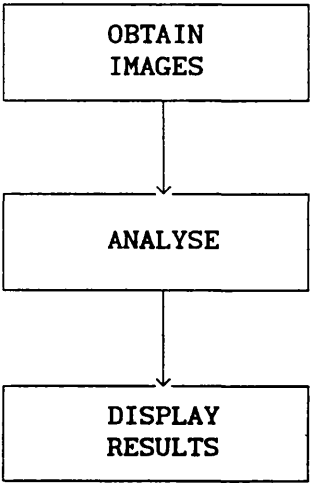


Fig. 7.11 The three basic stages of my image analysis, they can be regarded as three subtasks in parallel processing.

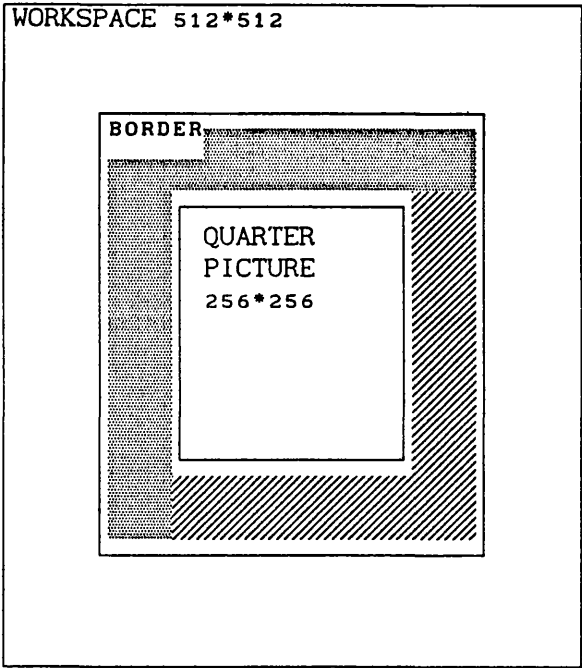




Fig. 7.12 The arrangement of workspace in each working transputer.

-  swapped edges
-  filled up edges



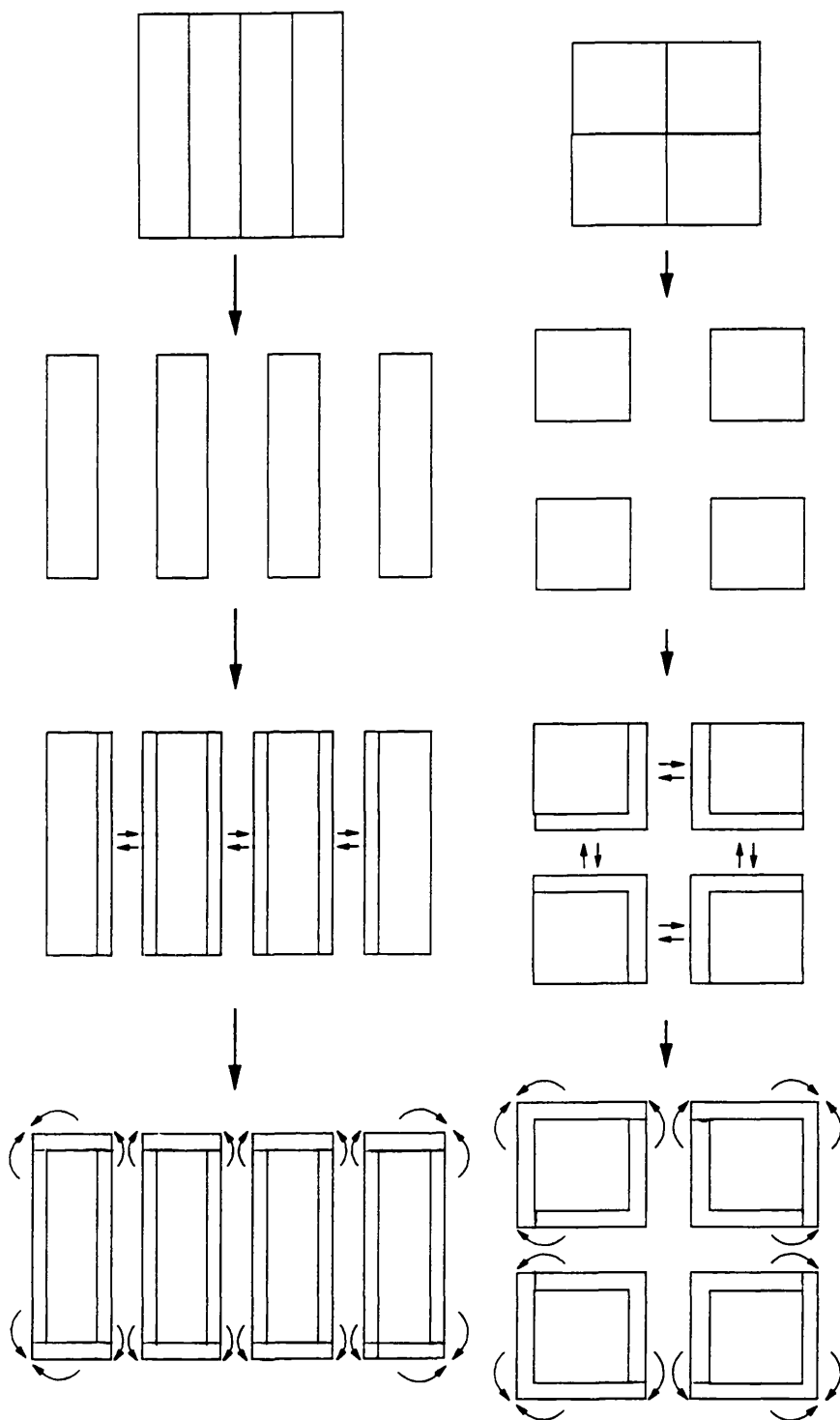


Fig. 7.13 Segmentation of an image onto four transputers.  
Right: tiles; Left: strips.  
Top: original image; up-middle: placements on the transputers;  
down-middle: swapping borders; bottom: filling edges.

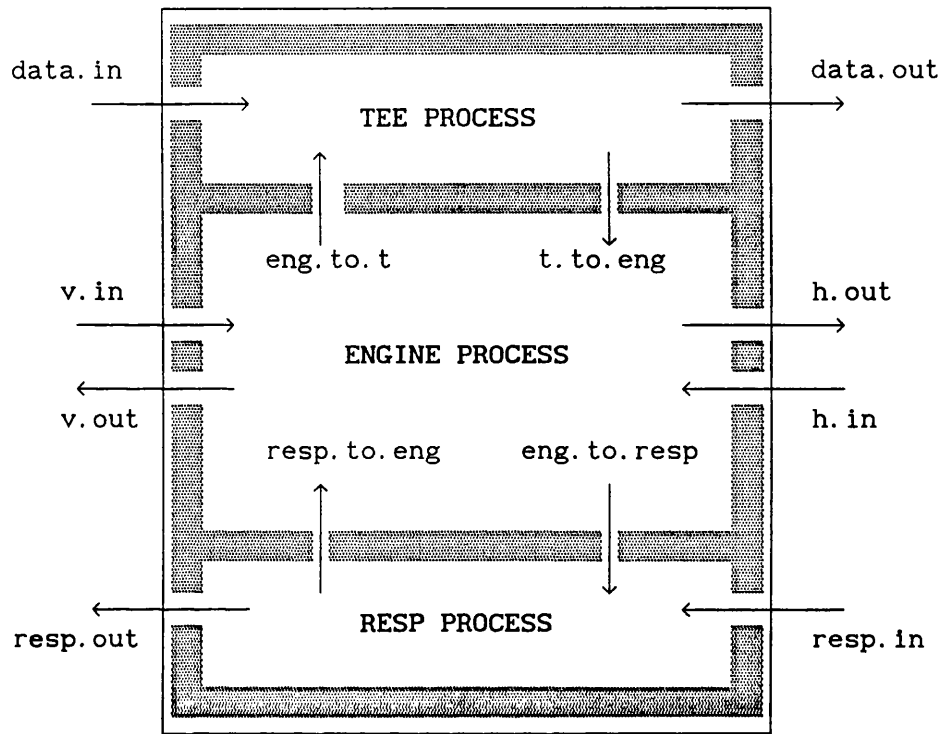
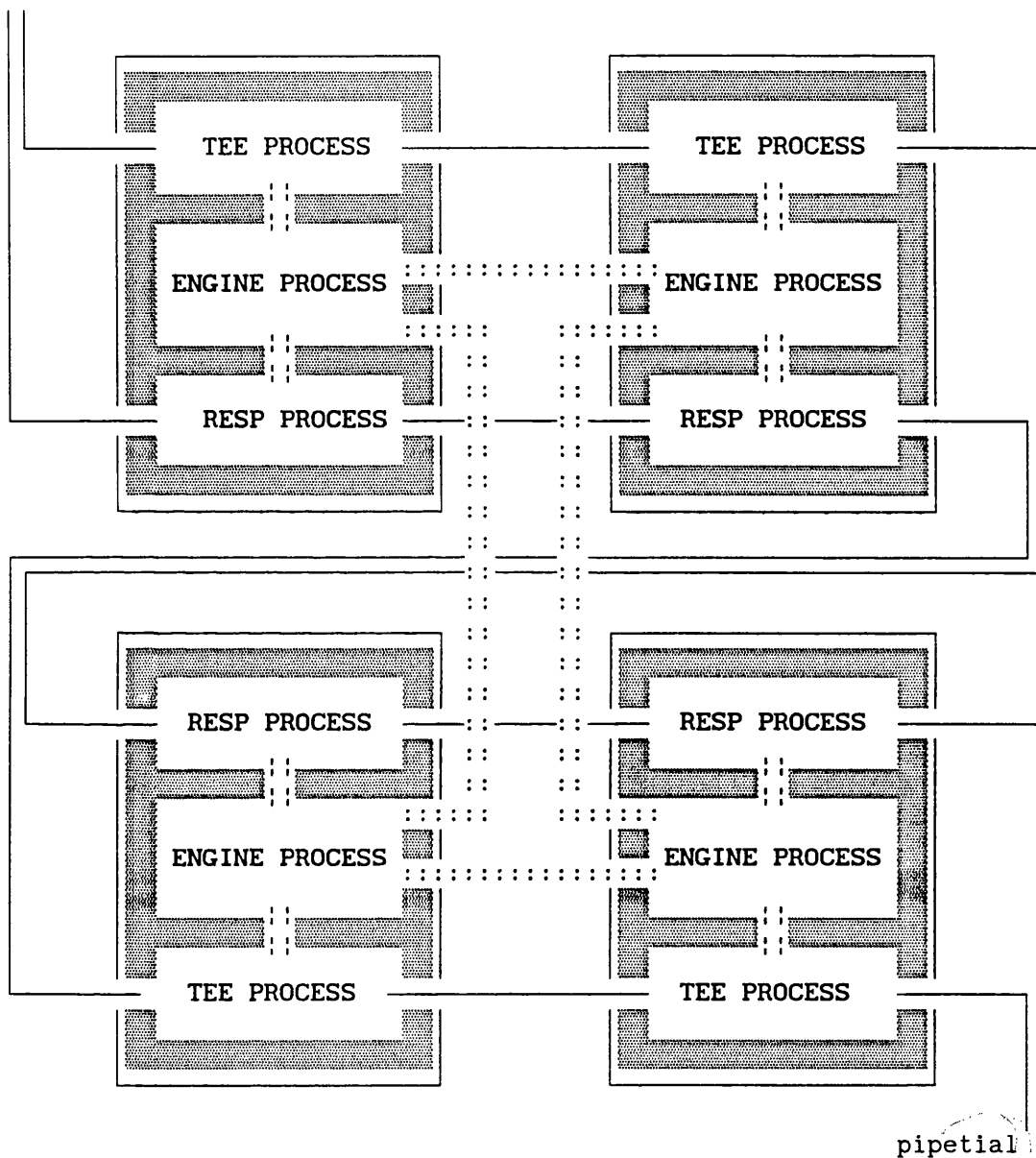


Fig. 7.14 Three processes on each working transputer running in parallel

pipehead



**Fig 7.15 Four working processors array on multiprocessor block**

== pipeline channels

```
===== internal channels
```

```
::::: swap-edge channels
```

HOST	MASTER	4 x WORKERS	HARLEQUIN
SEND DATA----	GET DATA	INITIALS	INITIALS
SEND ICON----	GET ICON	----	GET ICON
SEND DATA----	GET DATA		---->----DISPLAY D
SHOW H.D ----<----	FIND H.D	FIND U	---->----DISPLAY U
SEND CONT----	GET CONT		
SHOW H.U ----<----	ADD H.U	----<----	SEND H.U
SEND CONT----	SEND ICON	---->----	GET ICON
		FIND A,C	---->----DISPLAY A
SHOW H.A ----<----	ADD H.A	----<----	SEND H.A
SEND CONT----	PASS CONT	---->----	GET CONT
		SEND C	---->----DISPLAY C
		FIND T	---->----DISPLAY T
SHOW H.T ----<----	ADD H.T	----<----	SEND H.T
SEND CONT----	PASS CONT	---->----	GET CONT
		FIND B	---->----DISPLAY B
		FIND F	
SHOW F ----<----	ADD F	----<----	SEND F

Fig. 7.16 Simplified flow chart for Top-contouring mapping. D (=data), U, A, C, T, and B, are images; F is a vector (field data); H = histogram; ICON is a vector of control numbers; CONT is a command from the keyboard to continue. For fast use, some displays and histograms are omitted, and most CONTs originate from the master, these being used to ensure that a complete set of partial histograms are obtained and added together before the next set start to arrive. (Smart and Leng 1990).

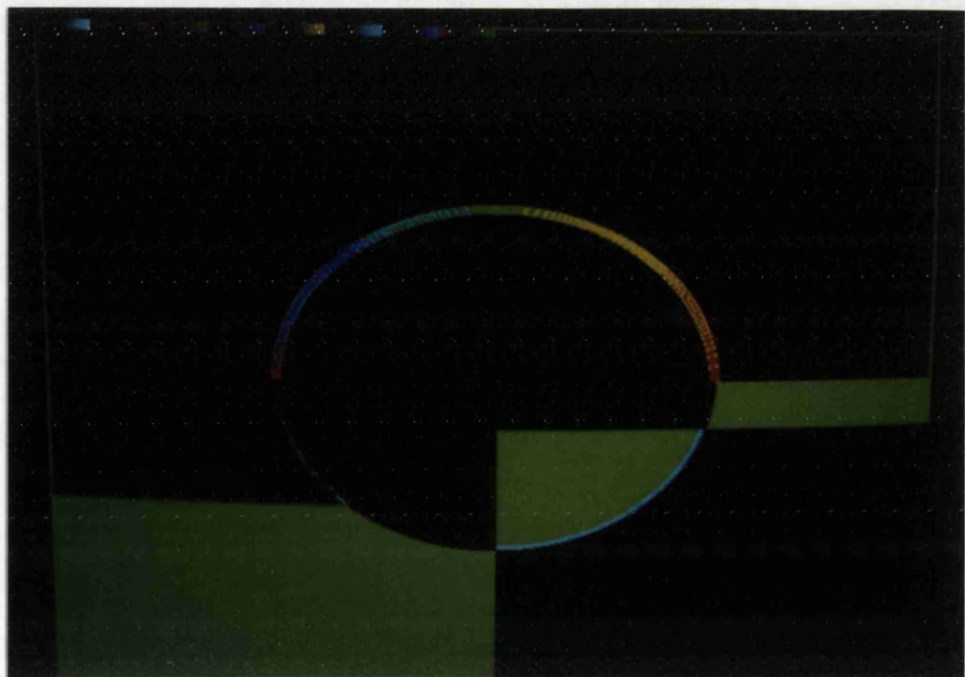


Fig. 7.17 Illustrating the use of four transputers to Top-Contour a synthetic image consisting of a single circle. The early version of the code has permitted the tile to be processed at different speeds; the top left is complete; the bottom right is a lot behind.

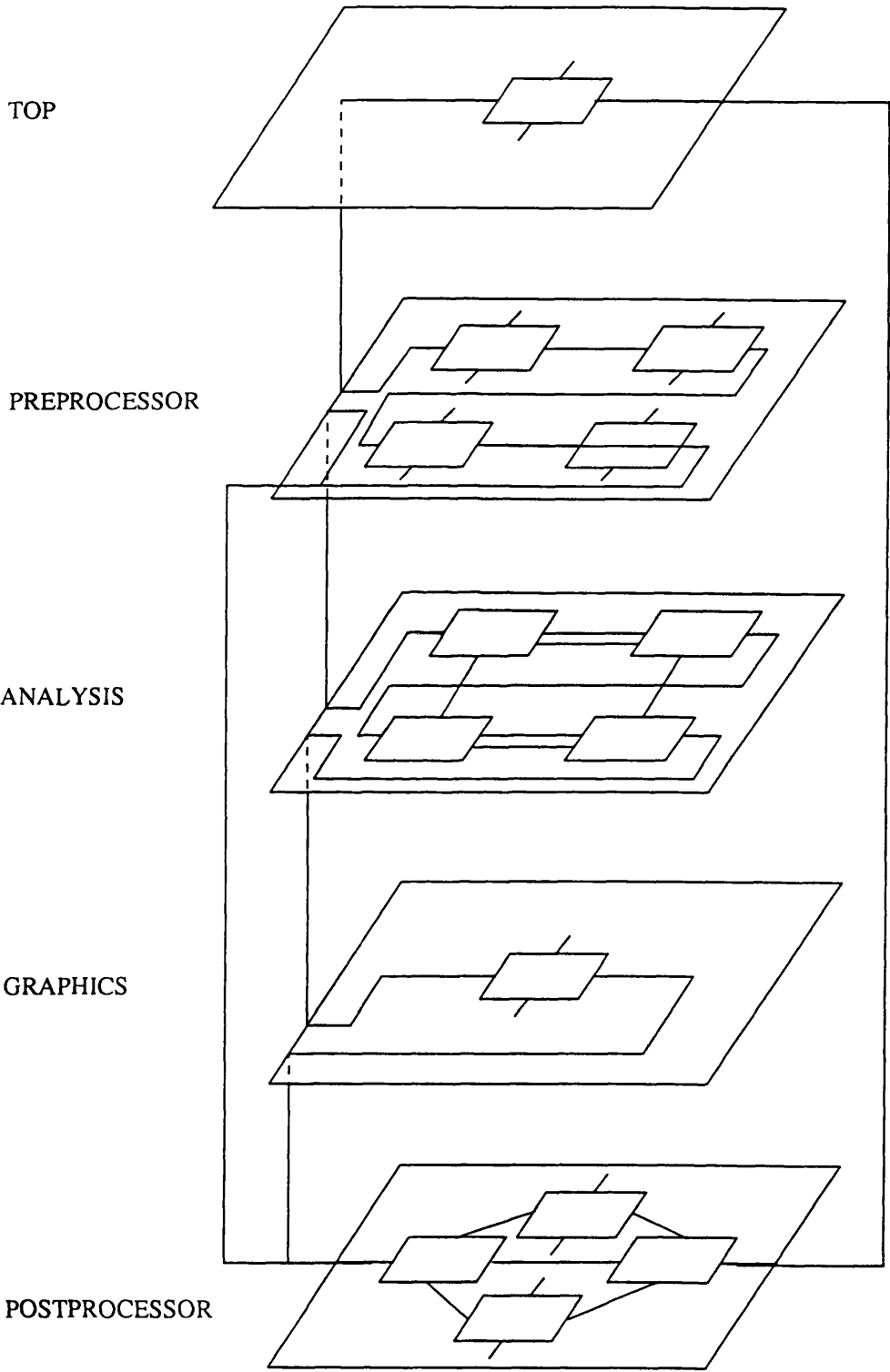


Fig. 7.18 Future parallel system structure.

## CHAPTER 8 CONCLUSIONS AND FUTURE WORK

In order to analyse textured images (such as scanning electron micrographs of clay), the study of this thesis covered both theoretical studies, development of software and the establishment of the hardware. The textural analysis in this work used the techniques of Image Segmentation, Image Transformation, and Mathematical Morphology. The textural analysis included analysis based on individual particles, groups of particles, and measurements taken over the whole image. The application of parallel processing in this research accelerated the calculations and allowed the development of more demanding algorithms.

### 8.1 CONCLUSIONS ON TEXTURAL ANALYSIS

#### 8.1.1 Analysis of Domain Texture

In scanning electron micrographs of clay, particles are normally lying in face-to-face groups, *i.e.* domains (Fig. 3.6). Additionally, ridges of mountains seen in aerial photographs sometimes all lie in the same direction (Fig. 3.15). Chapter 3 discussed two novel segmentations: Top-Contouring and Consistency Ratio Mapping. These two segmentations are based on the calculation of intensity gradient at each pixel in the image, then combine encoding of the angle (*i.e.* of the direction of the intensity gradient) and smoothing with a large

filter to obtain the local preferred orientation. They are used to map the areas which have the same local preferred orientation (called domains) or lack such orientation (called random clusters). These methods were shown to be very well suited for different scales of scanning and transmission electron micrographs, as can be seen in Figs. 3.1e, 3.7 and 3.14. They can also be used for analysing other kinds of images such as aerial photographs of islands (e.g. Fig. 3.15).

Consistency Ratio Mapping requires transputers with larger on-chip memory than Top-Contouring. It was developed after the calculations had been moved onto Glasgow University's MEIKO system, which made the required memory available. Consistency Ratio Mapping is more accurate than Top-Contouring Mapping. Consistency Ratio Mapping smooths before encoding, whereas Top-Contouring encodes before smoothing; and the present implementation of Consistency Ratio Mapping uses weighted smoothing rather than the unweighted smoothing used in Top-Contouring Mapping. However, it could also be possible to base Consistency Ratio Mapping on the unweighted smoothing *if appropriate*.

In future work, a further development of segmentation, which is suggested in Section 4.2.5, may be called Enhanced Orientation Mapping. This segmentation can selectively map the areas corresponding to the peaks of the polar histogram of the Enhanced Orientation Analysis. Thus this method draws attention on certain directions in which the majority of the particles lie, closely packed, with local preferred orientation stronger than other directions.

### 8.1.2 Analysis of Global Texture

Following Unitt's global orientation analysis, a comparison was made between weighted and unweighted polar histograms based on improved calculations of intensity gradient. It was shown that the weighted polar histogram may give a more distinct directional distribution. By using Enhanced Orientation Analysis explained in Chapter 4, the orientation information was enhanced, thereby giving a more obvious polar histogram and drawing attention to the dominant sub-parallel directions (these sub-parallel directions could be segmented by using Enhanced Orientation Mapping as noted above). Also in Chapter 4, a method of local porosity analysis was developed, which, for the first time, showed the relationship between the local porosity and the local orientation of the particles. This method of local porosity analysis is believed to be applicable to electron micrographs which are at a low magnification to give a large field of view and at too low a magnification to permit a 'black and white' segmentation of particles and voids.

This enhanced global orientation analysis was compared with the methods using the Hough transform, the Directed Vein Method, and a Convex Hull method. All three methods gave polar histograms similar to that of the Enhanced Orientation Analysis. In this work the slope-intercept Hough space was used taking precautions to avoid infinite slope and intercept when the line approaches a vertical position. To avoid the infinity problem, it is hoped that the  $\rho$ - $\theta$  Hough space could be used in future calculations.



### 8.1.3 Analysis of Individual Particle Texture

In order to get better results, the author suggested using skeletons of particles instead of edges in the Hough transform and Directed Vein methods.

Non-linear Greyscale Morphology was developed in Chapter 5 to analyse individual particle texture. This approach evades one of the most difficult stages in image processing, *i.e.* converting to a binary image. Although in the new method, a few parameters need to be set according to the characteristics of the image, the final result of successive cycles of erosion gave the skeletons of particles (Fig. 5.12). These skeletons could be used to analyse the orientation, separation, size distribution, and contact characteristics of individual particles in the image.

There are many methods which could be used to improve the skeletons of the images. The simplest method of cutting the joint lines in different directions is to measure whether there are more than two neighbouring points in the 3x3 mask, and they are not lying on the same direction, when the middle point is an on line pixel. A better method which could be used is the Hough transform to locate (and join) parts of broken lines in the same direction with small gaps in between and to cut contiguous non-colinear lines apart.

Some further applications of Non-linear Mathematical Morphology could be made. The immediate application of this new algorithm is to use the skeletons of particles for the Hough transform and the Method of Veins (Chapter 4) instead of using the edges of particles, and this

is expected to give a much more accurate result. Further, the skeletonized image could be used to count isolated points, the ends of particles, and the joints of particles, so as to provide measurements to classify the arrangement of particles in the images. Also, in some cases, the skeleton of the voids could be easily obtained to analyse the tortuosity of the voids.

In the course of these experiments, it was noted that, by taking a proper choice of parameters, the Non-linear Greyscale Morphology transform could produce a black-and-white image. This might be a new way of converting to a binary image.

#### 8.1.4 Use of The Semi-variogram

Both the analysis of domain texture and the analysis of global texture mentioned above used large uniform filters. No published method of deciding the size and shape of these filters was found. The introduction of the semi-variogram (Section 3.3.3) appears to have solved this problem and to be an important advance in the general practice of image analysis.

## 8.2 CONCLUSION ON PARALLEL PROCESSING APPLICATION

The latest development in parallel processing, the transputer, has provided a cheap, efficient, and flexible means of parallel computation. Transputers were employed in this research to implement the new textural analysis algorithms. One single and three multi-transputer systems were set up in turn to develop the algorithms. Because there was a limit on the SERC loan, a six

transputer system became the basic system for the calculations done in this research. In order to make full use of the transputer links and to build a suitable system for the textural analysis, the harness of the system was set up as a pipeline linking the six transputers one-by-one, with the four middle transputers also linked in an array. The top transputer controls the whole system and communicates with the host; the middle block of four transputers does the main calculations; and the bottom transputer mainly displays the result on a video screen. This system may be alternatively regarded as having three 'multi-processor' blocks in a pipeline.

In the image analysis performed in this research, three subtasks: obtaining the image, analysis, and displaying the result, have been located parallelly on the three multiprocessor blocks which correspond to the pipeline hardware. According to the requirement of the novel textural analyses of this work, data-structure parallelism was regarded as the most suitable parallelism for most of the calculations. Thus, each image data array was divided into four quarters and mapped onto the middle four transputers to do the calculations. When necessary, the inter-multiprocessor links, which connect the four transputers in the array, were used for data exchange during the calculations. This system made a powerful and convenient tool for the development of the new algorithms: the developer was able to concentrate on the kernel of the application, any necessary modifications to the harness being few and simple. In other words, the system was flexible, easy to program, and fast in execution.

Both the hardware links and the software configuration were designed to build an extensible system especially for the analysis of

the microstructure of scanning electron micrographs. Sets of images normally about 24 or even more, were taken from each mechanical test sample; some of the programs have been extended to analyse a whole set of up to 99 images automatically. The programs could be extended again to analyse all the samples from one experiment. Two further multiprocessor blocks may have to be installed in the present pipeline system, viz a preprocessing multi-transputer block before the analysis blocks, and a postprocessing multi-transputer block after the analysis block. Moreover, some electron microscope laboratories are planning to increase the image size from 512\*512 to 1024\*1024 pixels; and to follow this development, each multiprocessor block will then need 16 transputers to analyse 256\*256 size tiles on each transputer. Extensions such as these are easy to implement on transputer-based systems.

### 8.3 SUMMARY

The use of a set of six transputers provided an efficient means of developing and testing novel image analysis algorithms to analyse the texture of scanning electron micrographs of clay soils (and similar images). These algorithms have been taken into use in a cooperative study with the University of East Anglia and for further work in Glasgow University; and a derivative has been made available commercially in the Semper Image Analysis package.

## APPENDIX A CHOOSING THE NUMBER OF TRANSPUTERS

This appendix shows how there may be an optimum number of transputers for any particular task.

The calculation was made for the following assumptions:

Set up time using one transputer =  $s = 0.1$

Run time using one transputer =  $r = 0.9$

Total time using one transputer =  $r + s = 1.0$

Suppose there are  $w$  workers.

Additional set up time for first worker =  $s = 0.1$

Additional set up time for each extra worker =  $0.1s = 0.01$

Runtime =  $r/w$

Total time using  $w$  workers =  $r/w + 2s + (w-1)0.1s$

The results are summarized in Table A.1 and shown in Fig. A.1, where the larger gangs are assumed to contain a graphics transputer which does not count as a proper worker, but which still needs to be set up. For the case considered, the fastest combination of transputers had 9 workers in a gang of 11.

*In practice, the calculations should be based on measured component timings.*

Table A.1 Time for the multitransputer system (r = 0.9, s = 0.1).

No of transputers	No of workers	Formula	Time
1	1	$r + s$	1.0
2	1	$r + 2s$	1.1
3	1	$r + 2s + 0.01$	1.11
4	2	$r/2 + 2s + 0.02$	0.67
6	4	$r/4 + 2s + 0.04$	0.47
11	9	$r/9 + 2s + 0.09$	0.39
18	16	$r/16 + 2s + 0.16$	0.42
27	25	$r/25 + 2s + 0.25$	0.49
38	36	$r/36 + 2s + 0.36$	0.59
51	49	$r/49 + 2s + 0.49$	0.71
66	64	$r/64 + 2s + 0.64$	0.85

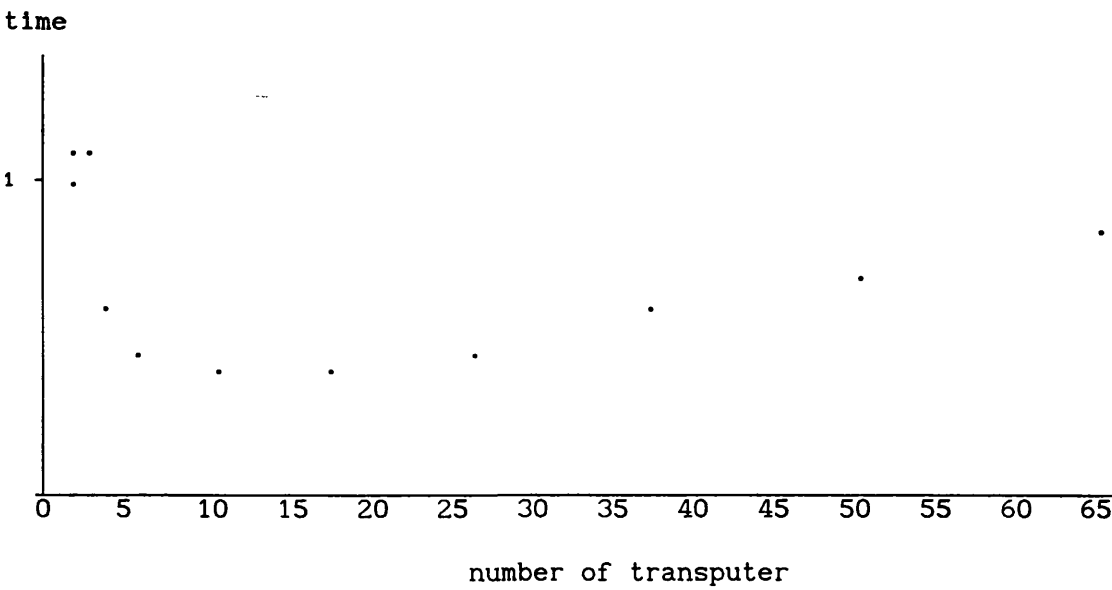


Fig A.1 Time vs number of transputers.

## APPENDIX B CODE FOR TEE JUNCTION PROCESS

This appendix contains the Occam code of the Tee Process, running on the working transputer, linking up the outwards directional pipe Fig. 7.14. The Tee Process behave like a gate. It receive data either from pipe-head to send downwards to pipe-tail and Engine Process; or from Engine Process downwards to pipe-tail. Whichever arrives first will be processed first, namely alternative structure (ALT). Anything passed into the Tee Process is passed out immediately, otherwise the Tee process will wait if it is not reading.

The Tee Process contains a Protocol and a reading process. The Protocol has all the data string formats listed. The reading process could be made to read in any kind of data string listed in the Protocol send to it at any time and then pass it on. However, in this version, histograms, his, and messages, mes, are ignored, because they usually travel in the inwards direction pipe. The details of what is passed are controlled by the Protocol. The destinations are controlled by the CASE statements. The code of the Tee Process is shown below.

PROTOCOL STD

CASE

```

    dat; INT; [512]BYTE
    ans; INT; INT; INT::[]BYTE
    his; INT::[]INT
    cmd; [4]BYTE; INT
    lut; INT
    mes; BYTE; INT; [12]BYTE; INT
    swi; INT; INT::[]INT
    swr; INT; INT::[]REAL32
    vec; INT::[]REAL32
    icn; [64]INT:

```

:

PROC tee.junction(CHAN OF STD data.in, data.out, t.to.eng, eng.to.t)

--- Xiaoling Leng & Peter Smart, Glasgow University

--- Version 92.11.23

```

[512]REAL32 vector:      -- for use if histo overflow
[512]INT histo:          -- histogram or other report
[512]BYTE line:          -- whole or part line of image
[64]INT icon:            -- control numbers
[12]BYTE message:        -- report from worker to screen
[4]BYTE comm:            -- command to another transputer
INT ii:                  -- number of first pixel
INT jj:                  -- number of line
INT length:              -- number of items sent
INT number:              -- control number, debug report, etc
INT member:              -- reference number of reporting transputers
BYTE gang:               -- reference letter of gang of transputers
BOOL working:

```

SEQ

working:=TRUE

WHILE working

ALT

data.in ? CASE

dat; jj; line

SEQ

data.out ! dat; jj; line

t.to.eng ! dat; jj; line

ans; ii; jj; length::line

data.out ! ans; ii; jj; length::line

his; length::histo

data.out ! his; length::histo

cmd; comm; number

SEQ

data.out ! cmd; comm; number

t.to.eng ! cmd; comm; number

lut; number

data.out ! lut; number

mes; gang; member; message; number

data.out ! mes; gang; member; message; number

swi; jj; length::histo

t.to.eng ! swi; jj; length::histo

swr; jj; length::vector

t.to.eng ! swr; jj; length::vector

vec; length::vector



```
        data.out ! vec; length::vector
    icn; icon
    SEQ
        data.out ! icn; icon
        t.to.eng ! icn; icon

eng.to.t ? CASE
    ans; ii; jj; length::line
        data.out ! ans; ii; jj; length::line
    his; length::histo
        data.out ! his; length::histo
    cmd; comm; number
        data.out ! cmd; comm; number
    lut; number
        data.out ! lut; number
    mes; gang; member; message; number
        data.out ! mes; gang; member; message; number
    swi; jj; length::histo
        data.out ! swi; jj; length::histo
    swr; jj; length::vector
        data.out ! swr; jj; length::vector
    vec; length::vector
        data.out ! vec; length::vector
:
```

## REFERENCES

- Abutaleb A S, 1989, *Automatic Thresholding of Grey-level Pictures Using Two-Dimensional Entropy*, *Comp. Vis., Graph. and Image Proc.*, Vol. 47, pp 22-32.
- Aggarwal J K, 1977, *Computer Methods in Image Analysis*, IEEE Press
- Bates R H T and McDonnell M J, 1986, *Image Restoration and Reconstruction*, Oxford University Press, Oxford.
- BS 598: Part 105, 1990, *Sampling and examination of bituminous mixtures for roads and other paved areas. Methods of test for the determination of texture depth*, British Standards Institute, London.
- BS 6741: 1987, *Glossary Surface roughness terms*, British Standards Institute, London.
- Chandy K and Misra J, 1988, *Parallel Program Design -- A Foundation*, Addison-Wesley Press.
- Connors R and Harlow C, 1980, *A theoretical Comparison of Texture Algorithms*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No 3, pp 222.
- Cook R and Sandys-Renton J, 1991, *A parallel segmentation algorithm (Merge Using Moments) for SAR images*, *Proc. 3rd International Conference on Applications of Transputers*, Glasgow. IOS Press, Amsterdam Vol. 1 pp 311-316.
- Costa L, Leng X, Sandler M and Smart P, 1991a, *Semi-automated analysis of clay samples*, *Review of Scientific Instruments*, Vol. 62, pp 2163-2166.
- Costa L, Leng X, Sandler M and Smart P, 1991b, *Analysis of clay microstructure by transputers*, *Proc. 3rd International Conference on Applications of Transputers*, Glasgow. IOS Press, Amsterdam. Vol. 1 pp 317-322.
- Costa L and Sandler M, 1990, *Implementation of the Binary Hough Transform in Pipelined Multi-Transputer Architectures*, in Pritchard D J and Scott C J eds *Applications of Transputer 2*, IOS Press, pp 150-155.
- Crichlow J M, 1988, *An Introduction to Distributed and Parallel Computing*, Prentice Hall Press.

- Curry J R, 1956, *The Analysis of Two-Dimensional Orientation Data*, J. of Geology, Vol. 64, pp 117-131.
- Dyer C R and Rosenfeld A, 1979, *Thinning Algorithms for Gray-Scale Pictures*, IEEE Trans. Pattern Analysis and Machine Intelligence Vol. PAMI-1 No. 1 pp 88-89.
- Eichmann G and Kasparis T, 1988, *Topologically Invariant Texture Descriptors*, Computer Vision Graphics and Image Processing Vol. 41 pp 267-281.
- Gonzalez R C and Wintz P, 1987, *Digital Image Processing -- Second Edition*, Addison-Wesley Press, Reading, Mass.
- Grant C D, Dexter A R and Huang C, 1990, *Roughness of soil fracture surfaces as a measure of soil microstructure*, J. Soil Science 41, pp 95-110.
- Haralick R M, Shanmugam K and Dinstein I, 1973, *Textural Features for Image Classification*, IEEE Trans. on Syst. Man, Cybern. SMC-6, pp 610-621.
- Haralick R M, 1986, *Statistical Image Texture Analysis*, in Young T Y and Fu K eds. *Handbook of Pattern Recognition and Image Processing*, Academic Press Inc Chapter 11 pp 247-279.
- Haralick R M, Sternberg S R and Zhuang X, 1987, *Image Analysis Using Mathematical Morphology*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No 4, pp 532-550.
- Hawkes P W et al, 1988, *Image and Signal Processing in Electron Microscopy*, Scanning Microscopy International
- Heijmans H J A M and Ronse C, 1990, *The Algebraic Basis of Mathematical Morphology 1, Dilations and Erosions*, Computer Vision, Graphics, and Image Processing, Vol. 50, pp 245-295.
- Hockney R W and Jesshope C R, 1988, *Parallel Computers 2 architecture, programming and algorithms*, Adam Hilger.
- Hough P V C, 1962, *Methods and Means for Recognising Complex Pattern*, U S Patent 3069654, known from Gonzales & Wintz 1987.
- INMOS Ltd, 1987, *The Transputer Family 1987*, INMOS Ltd.
- 1988a, *OCCAM 2 Reference Manual*, Prentice Hall.
- 1988b, *Transputer Development System*, Prentice Hall.
- 1988c, *Transputer Reference Manual*, Prentice Hall.
- 1988d, *IMS B008 User Guide and Reference Manual*, INMOS Ltd.
- 1989a, *The Transputer Databook*, second edition, INMOS Ltd.
- 1989b, *The Transputer Development and iQ Systems Databook*, INMOS Ltd.
- Jacobus C and Chien R T, 1981, *Intermediate-Level Vision Building -*

- Building Vertex-String-Surface (V-S-S) Graphs*, Computer Graphics and Image Processing, Vol. 15, No. 4, pp 339-363.
- Jardine L F and Whitworth I R, 1991, *Parallel Transputer Texture Analysis*, IEEE Colloquium on Parallel Architectures for Image Processing Applications pp. 11.1-11.5.
- Jones G and Goldsmith M, 1988, *Programming in OCCAM 2*, Prentice Hall.
- Joyce-Loebl Ltd, 1989, *Image Analysis -- Principles and Practice*, Joyce-Loebl Ltd.
- Krishnamurthy E V, 1989, *Parallel Processing -- Principles and Practice*, Addison-Wesley Press.
- Krumbein W C, 1939, *Preferred Orientation of Pebbles in Sedimentary Deposits*, J. of Geology, Vol. 47, pp 673-706.
- Leng X and Smart P, 1991, *Improved methods of textural analysis*, Proc. 3rd International Conference on Applications of Transputers, Glasgow. IOS Press, Amsterdam, pp 323-328.
- Leng X and Smart P, 1992, *Image Analysis of Clay Scanning Electron Micrographs*, Proc. 13th International Congress on X-ray optics and microanalysis, Manchester. Accepted.
- Levi G and Montanari U, 1970, *A Gray-Weighted Skeleton*, Inform. Contr., Vol. 17, pp 62-91.
- Levine M D, 1985, *Chapter 9 Texture*, pp. 422-479 in *Vision in Man and Machine*. McGraw-Hill.
- Mandelbrot B B, 1983, *The fractal geometry of nature*, W H Freeman, New York.
- Mardia K V, 1972, *Statistics of Directional Data*, Academic Press, London.
- Meyer F, 1978, *Contrast Feature Extraction*, in *Quantitative Analysis of Microstructures In Material Science Biology and Medicine*, ed. by J L Chermant, pp 374-380, Reder-Verlag, Stuttgart
- Nakayawa Y and Rosenfeld A, 1978, *A Note on the Use of Local Min and Max Operation in Digital Picture Processing*, IEEE Trans. Syst, Man, Cybern, Vol. SMC-8, pp 632-638.
- O'Brien N R and Slatt R M, 1990, *Argillareous Rock Atlas*, Springer-Verlag.
- Peley S and Rosenfeld A, 1981, *A Min-Max Medial Axis Transformation*, IEEE Trans Pattern Analysis and Machine Intelligence, Vol. PAMI-3, pp 208-210.
- Pentland A P, 1984, *Fractal-based description of natural scenes*, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 6, pp 661-674.

- Reiche P, 1938, *Analysis of Cross-lamination: The Coconino Sandstone*, J. of Geology, Vol. 46, pp 905-932.
- Rosenfeld A and Kak A C, 1976, *Digital Picture Processing*, Academic Press.
- Salari E and Siy P, 1984, *The Ridge-Seeking Method for Obtaining the Skeleton of Digital Images*, IEEE Tran. on Syst., Man, and Cyb., Vol. SMC-14, No 3, pp 524-528.
- Serra J, 1982, *Image Analysis and Mathematical Morphology*, Academic Press.
- Serra J, 1986, *Introduction to Mathematical Morphology*, Computer Vision, Graphics, and Image Processing, Vol. 35, pp 283-305.
- Shannon C E, and Weaver W, 1949, *The mathematical theory of communication*, University of Illinois Press, Urbana.
- Shih F Y C and Mitchell O M, 1991, *Decomposition of Grey-scale Morphological Structuring Elements*, Pattern Recognition, Vol. 24, No 3, pp 195-203.
- Sternberg S R, 1986, *Grayscale Morphology*, Computer Vision, Graphics, and Image Processing, Vol. 35, pp 333-355.
- Smart P and Leng X, 1990, *Image Analysis by Transputer*, in Pritchard D J and Scott C J eds. *Applications of Transputers 2*, Proc. 2 Int. Conf. Applications of Transputers, Southampton, IOS Press, Amsterdam, pp 240-247.
- Smart P and Leng X, 1991, *Textural analysis by transputer: Report on SERC Transputer Loan TR1/099*, SERC/DT1 Transputer Initiative Loan Reports Vol. 10, Ruthford Appleton Lab, Didcot, pp 1-15.
- Smart P and Leng X, 1992a, *Textural Analysis*, in Webber H C ed. *Image Processing and Transputers*, IOS Press, Amsterdam, Chapter 4 pp 73-95.
- Smart P and Leng X, 1992b, *Present Developments in Image Analysis*, invited review presentation for the Conference of Scanning Microscopy, Chicago
- Smart P, Leng X, and Awolaye O A, 1991, *Processing to the edges of images*, Report CE-MS-91-06, Civil Engineering Department, Glasgow University.
- Smart P and Tovey N K, 1981, *Electron microscopy of soils and sediments: examples*, Oxford University Press, Oxford.
- Smart P and Tovey N K, 1982, *Electron microscopy of soils and sediments: techniques*, Oxford University Press, Oxford.
- Smart P, Tovey N K, Leng X, Hounslow M W and McConnochie I, 1991. *Automatic analysis of microstructure of cohesive sediments*, pp

- 359-366 in Bennett R H, Bryant W R and Hulbert M H, eds., *Microstructure of fine-grained sediments: from mud to shale* Springer, New York. 1991.
- Smart P and Tovey N K, 1988, *Theoretical Aspects of Intensity Gradient Analysis*, Scanning, Vol. 10, pp 115-121.
- Snedecor G W, 1957, *Statistical methods applied to experiments in agriculture and biology*, Iowa State University Press, Ames.
- Sternberg S R, 1982, *Cellular Computer and Biomedical Image Processing in Biomedical and Computer*, Ed by Sklansky J & Bisconte J C, *Lecture Notes in Medical / Informatics*, Vol. 17, pp 294-319, Springer-Verlag, Berlin
- Thomas W, 1973, *SPSE Handbook of Photographic Science and Engineering*, Wiley, New York.
- Tovey K N and Smart P, 1986, *Intensity Gradient Techniques for Orientation Analysis of Electron Micrographs*, Scanning Vol. 8, pp 75-92.
- Tovey K N and Sokolov, 1981, *Quantitative SEM Methods for Soil Fabric Analysis*, Scanning Electron Microscopy.
- Transtech, 1988, *Transtech TSB07 Frame Grabber / Graphics Board for the IBM-User Manual*, Transtech Ltd.
- Unitt B M, 1975, *A digital Computer Method for Revealing Directional Information in Images*, J. Phys. E. Scientific Instruments, Vol. 8, pp 423-425.
- Unitt B M and Smith K, 1976, *The Application of the Mini Computer in Scanning Eletron Microscopy*, in D.G. Brandon ed *Electron Microscopy*, Vol. 1. TAL International Publishing Company. pp 162-167.
- Vilnrotter F M, Nevatia R and Price K E, 1986, *Structural analysis of natural textures*, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 pp 76-89.
- Webster R and Oliver M A, 1990, *Statistical methods in soil and land resource survey*, Oxford University Press, Oxford.
- Wegman E J and DePriest D J, 1986, *Statistical Image Processing and Graphics*, Marcel Dekker Inc.
- Wilson R and Spann M, 1988, *Image Segmentation and Uncertainty*, Research Studies Press Ltd.